

**Optimum Sensor Localization/Selection in
A Diagnostic/Prognostic Architecture**

A Dissertation

Presented to

The Academic Faculty

By

Guangfan Zhang

In Partial Fulfillment

Of the requirements for the Degree

Doctor of Philosophy in Electrical Engineering

Georgia Institute of Technology

January 2005

Optimum Sensor Localization/Selection in A Diagnostic/Prognostic Architecture

Approved by:

Dr. Bonnie S. Heck, Committee Chair

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. Thomas E. Michaels

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. Aldo A. Ferri

School of Mechanical Engineering

Georgia Institute of Technology

Dr. George J. Vachtsevanos

School of Electrical & Computer Engineering

Georgia Institute of Technology

Dr. A. Bruno Frazier

School of Electrical & Computer Engineering

Georgia Institute of Technology

Date Approved: January 2005

**TO MY PARENTS, MY WIFE, AND MY DAUGHTER,
WHO MADE THE JOURNEY ENJOYABLE**

ACKNOWLEDGEMENT

First of all, I would like to express my special and sincere gratitude to my advisor Dr. George Vachtsevanos for his persistent guidance and trust throughout my research. His enthusiasm for career affected me greatly.

Secondly, I wish to thank all my committee members, especially Drs. Bonnie S. Heck, Dr. Thomas Michaels for serving my reading committee. I thank them for their time, support, and suggestions.

I thank all present and past members of the Intelligent Control Systems Laboratory, whose assistance, encouragement, and friendship made the lab a lovely place, especially people in the PEDS group: Dr. Nicholas Propes, Dr. Seungkoo Lee, Dr. Yongsheng Zhao, and Dr. Irtaza Barlas. Special thanks must be given to Dr. Yuhua Ding, Dr. Lichu Zhao, Dr. Biqing Wu, and Dr. Liang Tang for their sincere friendship, warm-hearted and on-demand help to me and to my family, and technical and moral support throughout the research.

I am forever indebted to my parents, Wenying Lu and Zhaoyang Zhang, for constantly providing me the moral support and encouragement, which enabled me to achieve every goal in my life. I thank my elder sister, Yongping Zhang, my brother-in-law, Hugh Iwanicki, my Sister-in-law, Yingqi Zhang, my friends who always support my research and my life.

Finally, I am deeply grateful to my wife, Yingchuan Zhang, who shared with me the good times and supported me through the bad times with her love, encouragement,

incredible understanding, great tolerance, and sacrifices. Special thanks to my daughter, Cheryl, who brought me joy everyday and made even the hardships enjoyable.

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	iv
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
GLOSSARY	xi
SUMMARY	xiii
CHAPTER	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Definition	2
1.3 Assumptions	3
2 BACKGROUND.....	4
2.1 Integrated Diagnostic/Prognostic Architecture.....	5
2.1.1 Integrated Diagnostic/Prognostic Algorithms	5
2.1.2 Integrated Diagnostic/Prognostic Architecture	7
2.1.3 Open Questions	11
2.2 Optimum Sensor Localization/Selection (SLS).....	11
2.2.1 Component Level Sensor Localization	15
2.2.2 System Level Sensor Localization/Selection	17
2.2.3 Sensor Uncertainty	22
2.2.3 Open Questions	29
2.3 Research Focus.....	30
3 INTEGRATED DIAGNOSTIC/PROGNOSTIC ARCHITECTURE	31
3.1 Subsystem Modules	33

3.2 Software Architecture	34
3.3 COM/DCOM.....	36
3.4 Database Management	36
3.5 Graphical User Interface (GUI) and Communication	37
4 SENSOR LOCALIZATION/SELECTION FOR FAULT DIAGNOSIS	39
4.1 Sensor Localization/Selection Architecture	39
4.2 Requirements Analysis.....	41
4.2.1 Detectability	42
4.2.2 Uncertainty Issues in Detectability	47
4.2.3 Identifiability.....	49
4.2.4 Fault Detection Reliability (R).....	51
4.2.5 Limited Resources.....	52
4.3 FMECA Study	52
4.4 The Modeling Approach.....	53
4.5 FOM	57
4.6 Optimization	60
4.7 Performance Assessment.....	66
5 APPLICATION EXAMPLES	69
5.1 Process Demonstrator	69
5.1.1 Problem Statement	71
5.1.2 Diagnostic/Prognostic Results.....	72
5.1.3 Diagnostic Performance Assessment	75
5.1.4 Prognostic Performance Assessment.....	78
5.2 Optimum Sensor Placement	82
5.2.1 Problem Statement	82
5.2.2 Results Based on QDG model.....	87

5.2.3 Results Considering Detectability Uncertainty	90
5.2.3 Performance Assessment.....	92
5.2.4 Performance Assessment Feedback	95
6 CONCLUSIONS & CONTRIBUTIONS.....	98
6.1 Conclusions	98
6.2 Significant Contributions.....	99
6.3 Future Work.....	100
REFERENCES.....	101
APPENDIX A: FLOW CHART FOR GREEDY SEARCH ALGORITHMS II TO SOLVE OBSERVABILITY [9]	109
APPENDIX B: FLOW CHART FOR GREEDY SEARCH ALGORITHM III TO ACHIEVE MAXIMUM FAULT RESOLUTION [9]	110
APPENDIX C: LIST OF PUBLICATIONS.....	111
VITA	112

LIST OF TABLES

TABLE

1: REVIEW OF SENSOR PLACEMENT METHODS [3].	13
2: FAILURE MODES, OPERATING MODES, AND FEATURES.	72
3: FAULT BELIEF OF TANK1 LEAKAGE.	74
4: PROGNOSTIC RESULTS ANALYSIS OF TANK 1 LEAKAGE.	74
5: FUZZY RESULTS.	76
6: WNN RESULTS.	76
7: AFNN RESULTS.	77
8: DEMPSTER-SHAFER RESULTS.	77
9: RISE TIME AND SETTLING TIME ANALYSIS.	80
10: ACCURACY RESULTS.	81
11: FAULTS & SENSORS.	84
12: FACTORS TO DETERMINE THE FAULT DETECTABILITY.	85
13: SENSORS INFORMATION.	86
14: FAULTS INFORMATION.	87
15: ANALYSIS OF RESULTS.	89
16: QDG METHOD CONSIDERING UNCERTAINTY.	91
17: SDG PERFORMANCE ANALYSIS.	93
18: QDG PERFORMANCE ANALYSIS.	94
19: PERFORMANCE COMPARISON.	95
20: RESULTS AFTER PERFORMANCE FEEDBACK.	96
21: RESULTS COMPARISON WITH PERFORMANCE FEEDBACK.	97

LIST OF FIGURES

FIGURE

1: TYPICAL DIAGNOSTIC/PROGNOSTIC ARCHITECTURE.	4
2: THE TRI-REASONER INTEGRATED VEHICLE.	9
3: ARCHITECTURE OVERVIEW.	10
4: “PLUG AND PLAY” TOASTER MODEL.	11
5: DIFFERENT VIEWS OF SENSOR PLACEMENT.	14
6: BASIC COMPONENTS OF SENSOR PLACEMENT FOR FAULT DIAGNOSIS.	15
7: CLOSED-LOOP CONTROL SYSTEM.	18
8: PARTICLE SWARM OPTIMIZATION.	21
9: INTEGRATED SYSTEM ARCHITECTURE FOR DIAGNOSTICS/PROGNOSTICS.	32
10: SOFTWARE ARCHITECTURE.	35
11: SENSOR LOCALIZATION/SELECTION HIERARCHICAL STRUCTURE.	39
12: SENSOR LOCALIZATION/SELECTION ARCHITECTURE.	40
13: THE RELATIONSHIPS OF DETECTABILITY AND THE AFFECTING FACTORS.	46
15: A QDG EXAMPLE.	56
16: PERFORMANCE ASSESSMENT FEEDBACK.	68
17: A PROCESS DEMONSTRATOR.	70
18: PROCESS DEMONSTRATOR DIAGRAM IN LABVIEW.	71
19: GRAPHICAL USER INTERFACE.	73
20: A DIAGNOSTIC RESULT EXAMPLE.	73
21: A PROGNOSTIC RESULT EXAMPLE.	75
22: DWNN PREDICTION RESULTS.	79
23: ACCURACY ANALYSIS.	82
24: A FIVE-TANK SYSTEM.	83
25: THE QDG MODEL OF THE FIVE-TANK SYSTEM.	88
26: DETECTABILITY COMPARISON CONSIDERING UNCERTAINTY.	92

GLOSSARY

Failure: is an unexpected behavior, deviation from the normal behaviors, negative effects to the system, major plan breakdowns, substantial material damage, or complete breakdown.

Fault: is a physical or operational indication of abnormality in the system that indicates an incipient failure.

FMECA: Failure Modes and Effects Criticality Analysis.

Fault (Failure) Detection: an abnormal operating condition is detected and reported.

Fault (Failure) Isolation: determining which component (subsystem, system) is failing or has failed.

Fault (Failure) Identification: estimating the extent of the fault (failure).

Diagnostics: detecting, isolating and identifying an impending or incipient failure condition.

Prognostics: predicts the time window over which maintenance must be performed without compromising the system's operational integrity.

CBM: condition-based maintenance, determines the "optimum" time to perform maintenance.

Verification: it answers the question: "Have I built the system right?" (that is, does the system as built meet the performance specifications as stated?)

Validation: it answers the question: “Have I built the right system?” (that is, is the system model close enough to the physical system and are the performance specifications and system constraints correct?)

Detectability: is the extent to which the diagnostic scheme can detect the presence of a particular fault; it relates to the smallest failure signature that can be detected and the percentage of false alarms

Identifiability: goes one step further in distinguishing between various failure modes, once the presence of a failure has been established. It targets questions such as the source, location, type, and consequence of a failure and the distinguishability between sensor, actuator or system component failures.

Quantified-Directed-Graph (QDG): a quantitative approach proposed to model fault propagation within a system’s range.

SUMMARY

This research addresses the problem of sensor localization/selection for fault diagnostic purposes in Prognostics and Health Management (PHM)/Condition-Based Maintenance (CBM) systems. The performance of PHM/CBM systems relies not only on the diagnostic/prognostic algorithms used, but also on the types, location, and number of sensors selected. Most of the research reported in the area of sensor localization/selection for fault diagnosis focuses on qualitative analysis and lacks a uniform figure of merit. Moreover, sensor localization/selection is mainly studied as an open-loop problem without considering the performance feedback from the on-line diagnostic/prognostic system. In this research, a novel approach for sensor localization/selection is proposed in an integrated diagnostic/prognostic architecture to achieve maximum diagnostic performance.

First, a fault detectability metric, which expresses the capability of a sensor to detect a fault, is defined quantitatively. A novel graph-based approach, the Quantified-Directed Model, is called upon to model fault propagation in complex systems and an appropriate figure-of-merit is defined to maximize fault detectability and minimize the required number of sensors while achieving optimum performance.

Secondly, the proposed sensor localization/selection strategy is integrated into a viable and cost-effective diagnostic/prognostic system architecture together with mode identification, diagnostics, and prognostics while exhibiting attributes of flexibility and scalability. Moreover, the performance of the sensor localization/selection strategy is validated and verified in the integrated diagnostic/prognostic architecture, and the

performance of the integrated diagnostic/prognostic architecture acts as useful feedback information for further optimizing the sensors considered. The approach is tested through a five-tank simulation system and is validated in the integrated diagnostic/prognostic architecture.

This research has led to the following major contributions:

- A generalized methodology for sensor localization/selection for fault diagnostic purposes.
- A quantitative definition of fault detection ability of a sensor, a novel Quantified-Directed Model (QDG) method for fault propagation modeling purposes, and a generalized figure of merit to maximize fault detectability and minimize the required number of sensors while achieving optimum diagnostic performance at the system level.
- A novel, integrated architecture for a diagnostic/prognostic system to integrate the functions of sensor localization/selection, feature extraction, mode identification, and fault diagnosis and prognosis.
- Validation of the proposed sensor localization/selection approach in the integrated diagnostic/prognostic architecture.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the industrial and manufacturing arenas, Prognostics and Health Management (PHM) is a complex task that involves finding the “optimum” time to perform maintenance within the window prescribed by prognostic algorithms while meeting a host of constraints. With advances in computing power, diagnostic/prognostic techniques are becoming more efficient in detecting the presence of a fault and predicting the remaining useful life time of a faulty component. Meanwhile, optimum type, number, and location of sensors (sensor localization/selection) improve the diagnostic and prognostic capabilities of PHM systems. However, few researchers have worked on the problem of how to find optimum types, location, and number of sensors for fault diagnosis. Therefore, the first motivation of this research is to study the sensor localization/selection for fault diagnostic purposes.

Previous PHM applications focus on either diagnostics or prognostics while assuming sensor information required by diagnostics and prognostics was always available. Even if sensor localization/selection was considered, the performance of the sensor localization/selection method was not validated and verified with diagnostic/prognostic results, and the available sensor localization/selection methodology was an open-loop process without considering on-line performance of fault detection.

To overcome these limitations, this research proposes a novel approach to address the sensor localization and selection problem for fault diagnostic purposes in a flexible/open/scalable diagnostic/prognostic architecture.

1.2 Problem Definition

In this dissertation, a novel approach to address sensor localization/selection problems in a diagnostic/prognostic architecture is developed. Specifically, the following areas are addressed:

- Define the ability of a sensor to detect a fault quantitatively. Many factors contribute to the detection ability, for example, Signal-to-Noise Ratio (SNR) and detection sensitivity. Factors such as these contribute to define sensor fault detectability.
- Propose a Quantified Directed Graph (QDG) method to model fault propagation and to calculate the sensor fault detectability. Previous research efforts, such as Directed-Graph (DG), Signed-Directed-Graph (SDG), and fault tree methods, primarily focused on model fault propagation qualitatively. These methods were unable to specify fault propagation information quantitatively using these models. In this dissertation, a QDG method is defined and utilized to analyze the sensor fault detectability quantitatively.
- Define a generalized Figure-Of-Merit (FOM) and optimize it using a particle swarm optimization method to achieve the optimal sensor localization/selection for fault diagnostic purposes.

- Propose and implement an integrated and generic architecture for diagnostic and prognostic systems to take advantage of the proposed sensor localization/selection approach as well as rapidly emerging diagnostic and prognostic algorithms.

The remainder of this dissertation includes five chapters. In Chapter 2, the state-of-the-art techniques are briefly reviewed. In Chapter 3, an integrated system architecture is proposed for sensor selection/localization, mode identification, diagnosis and prognosis. In Chapter 4, a novel sensor localization/selection approach for fault diagnosis at the system level is presented. In Chapter 5, the proposed architecture is illustrated in a laboratory process demonstrator, and the proposed sensor localization/selection approach is validated with a five-tank simulation system. Finally, the research is summarized and contributions are provided in the last chapter.

1.3 Assumptions

The following assumptions are applied to this research.

- A system is assumed to have multiple faults and failures with different symptoms.
- Failures are assumed to be progressive in nature; abrupt failures such as spikes or sudden breakdowns are not considered in this research.
- The term fault identification will be used interchangeably with fault isolation.
- A system can be approximated linearly within a relatively small dynamic range for the purpose of building a quantified-directed-graph model.

CHAPTER 2

BACKGROUND

A typical diagnostic/prognostic architecture is shown in Figure 1 [40]. It includes sensor/sensing, data preprocessing/management, feature extraction, diagnostics, prognostics, and Condition-Based Maintenance (CBM). Each module is critical to system performance. In this research, we will focus on only two topics: the diagnostic/prognostic architecture and sensor localization/selection for fault diagnosis of complex systems.

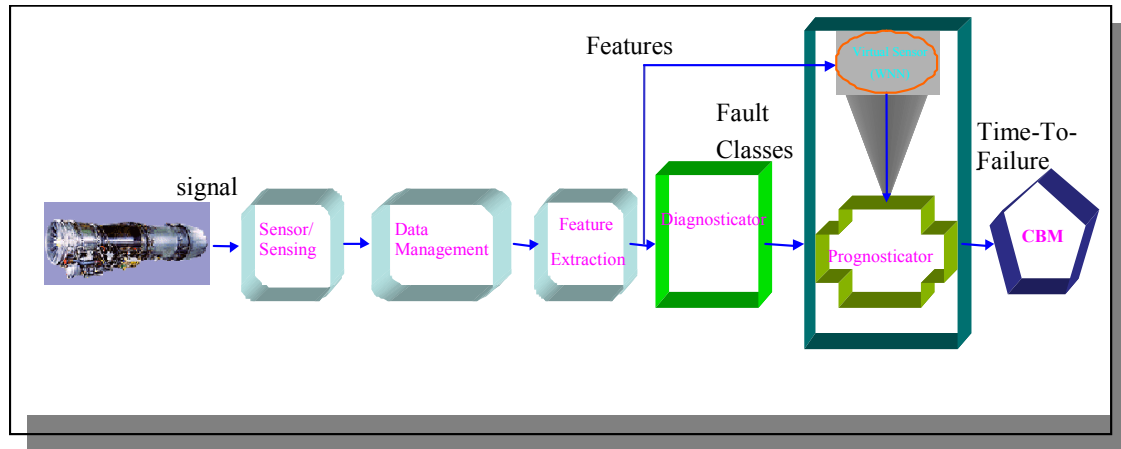


Figure 1: Typical diagnostic/prognostic architecture.

2.1 Integrated Diagnostic/Prognostic Architecture

The objective of machine health management is to diagnose a fault (incipient failure) as early as possible and to prognose the remaining useful lifetime of the faulty component. In this section, relevant work on the diagnostic/prognostic algorithms and the associated architecture are summarized.

2.1.1 Integrated Diagnostic/Prognostic Algorithms

A large number of approaches to diagnostics/prognostics have been reported in the technical literature. Diagnostics, or Fault Detection and Identification (FDI), attempts to recognize impending or incipient failures in processes and systems and forms a solid basis for Condition-Based Maintenance (CBM). Fault diagnosis is a relatively mature field with contributions ranging from model-based techniques to data-driven configurations that capitalize on soft computing and other “intelligent” tools [27][28]. Model-based techniques utilize a physical model and require a detailed and thorough understanding of the system [32]. As manufacturing facilities become complex and highly sophisticated, they are characterized by highly nonlinear dynamics coupling a variety of physical phenomena in the temporal and spatial domains. It is not surprising, therefore, that these processes are not well understood, making it difficult to build a precise model [29]. In a data-driven model, data is analyzed and used directly by various classification tools, such as Fuzzy Logic and Artificial Neural Networks (ANN) [33], [34], [35]. Some of the techniques can also be combined together to achieve higher fault diagnostic accuracy. A method of combining the discrete cosine transform technique with neural networks was considered to identify orbits of shaft centerlines of rotating machinery [36]. A wavelet transform and ANNs were combined and applied to

machinery fault diagnosis [37][38]. An integrated diagnostic algorithm, which combined Wavelet Neural Network (WNN) for the classification of high-frequency data and fuzzy logic for classification of low frequency data, was addressed in [39]. Combining the results from WNN and fuzzy logic using the Dempster-Shafer theory, the integrated diagnostic algorithm is able to identify both the low-frequency process faults and high-frequency faults.

In the industrial and manufacturing arenas, prognosis answers the question: what is the remaining useful lifetime of a machine or a component once an impending failure condition is detected and identified? During the past years, prognostic algorithms were developed based on stochastic models [41], knowledge-intensive expert systems [42], polynomial neural networks [43] and other techniques. However, these methods have yet to produce a systematic, efficient and robust approach to the prognostic problems. More recently, two main approaches have emerged as potential candidates for prognosis [44][59]. The first one relies on system models and state estimation techniques (Kalman and Alpha-Beta-Gamma tracking filters) to determine the remaining useful lifetime. The second uses a feature extractor and a learned association method, typically a neural network construct. The first category is hampered by the need for accurate system models while the second requires a sufficient database that covers the dynamic range of the machine or process for training and validation purposes. Within the second category, Dynamic Wavelet Neural Networks (DWNN) [12] incorporate temporal information and storage capacity into their functionality so that it can predict into the future and carry out fault prognostic tasks. An example [12] was presented where a trained static wavelet

neural network and a dynamic wavelet neural network successfully diagnose and prognose a defective bearing with a crack in its inner race.

In addition to predicting the remaining useful lifetime, it is crucial to assess the confidence of the prediction as well [58]. Over the past years, confidence interval estimation using sophisticated techniques has been developed. Some of these techniques are found in the classical forecasting literature [45][46][47], while others are devised for modern prediction models such as ANNs [48][49][50]. Unlike the traditional model-based methods, ANNs are data-driven and self-adaptive and make very few assumptions about the models for problems under study. ANNs learn from examples and capture the subtle functional relationship among the data. Thus, ANNs are well suited for practical problems where it is easier to have data than to have knowledge governing the system being studied. Werbos [52] reported that ANNs trained with the back propagation algorithm outperformed the traditional statistical methods such as the regression and Box-Jenkins approaches. The Confidence Prediction Neural Network (CPNN) was designed based on the General Regression Neural Network as a universal approximator for smooth functions [51]. It was built upon the features of its predecessors and employed a confidence distribution approximator node in its structure to represent uncertainty in the form of a confidence distribution.

2.1.2 Integrated Diagnostic/Prognostic Architecture

An integrated diagnostic/prognostic architecture is required to provide better information for condition-based maintenance approaches.

Su *et al.* [13] presented a generic prognostic framework by using model-based reasoning to integrate embedded test and sensor data into diagnostic and prognostic

information. Phoha [17] established distributed system architectures for electronic delivery of on-line equipment Monitoring, Diagnostics and Prognostics (MD&P) services. Roemer and his coworkers [18] integrated component, subsystem, and system level health monitoring strategies, with a modeling architecture that addresses failure mode mitigation and life cycle costs. Keller *et al.* [60] introduced a flexible and extensible architecture that directly supported the implementation of Integrated Vehicle Health Management (IVHM) systems. IVHM systems provided a reduction in the amount of data traffic required among systems, more accurate diagnostic capability, the addition of a prognostic capability, and lower support costs. The IVHM architecture consisted of six layers that spanned the processing of sensor data through decision support: signal processing, condition monitoring, health assessment, prognostics, decision support, and presentation.

Atlas and his coworkers discussed an evolvable tri-reasoner for IVHM [16] systems, as shown in Figure 2. In this architecture, an anomaly reasoner, a diagnostic reasoner, and prognostic reasoner were integrated to monitor the vehicle's health.

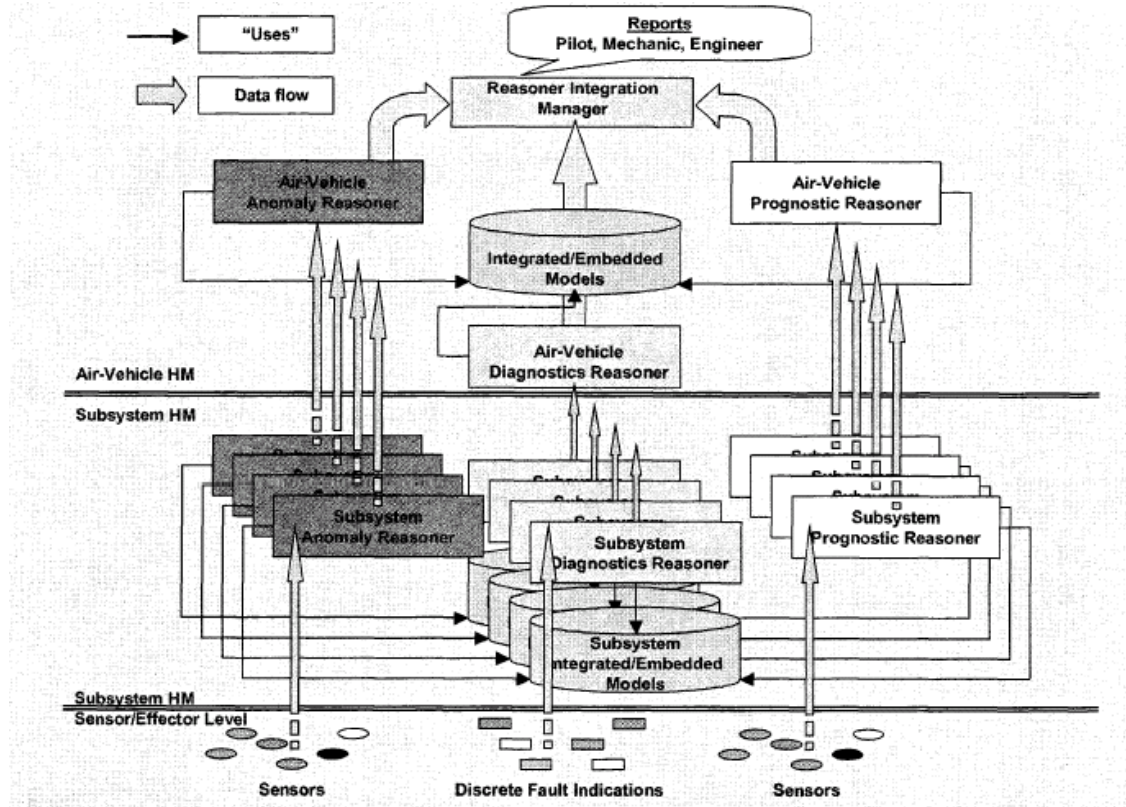


Figure 2: The tri-reasoner integrated vehicle.

Saeks [57] designed a hybrid supervised/unsupervised neural network architecture that was able to detect faults and fault precursors that were not included in the diagnostic database used to train the neural network (Figure 3).

Carl and his coworkers presented a “toaster” model [15] to illustrate the concept for “plug and play” functionality of the modules (Figure 4). In this architecture, each module, including anomaly detection, diagnostics, prognostics, human system interface, and others were integrated in an open system architecture. Prognostics was regarded as horizontal or vertical modules in this architecture. A horizontal module used anomaly detection and failure mode diagnosis information to make a prognosis. This was usually a

more accurate method, in which the knowledge about the type of failure and its severity was integrated into the time series prediction. By contrast, a vertical prognostic module was not explicitly dependent on the diagnostic information and its inputs consisted of either time and usage conditions or some additional measured data. For instance, experience-based statistical failure distributions can be applied to determine the probability of failure within a future time period given the prior time/usage history.

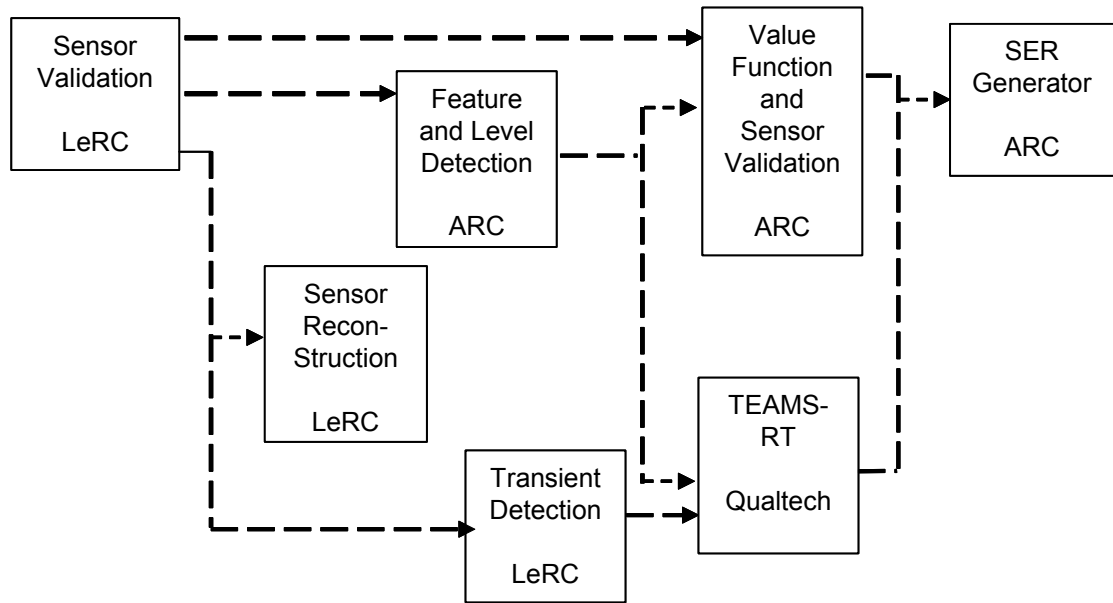


Figure 3: Architecture overview.

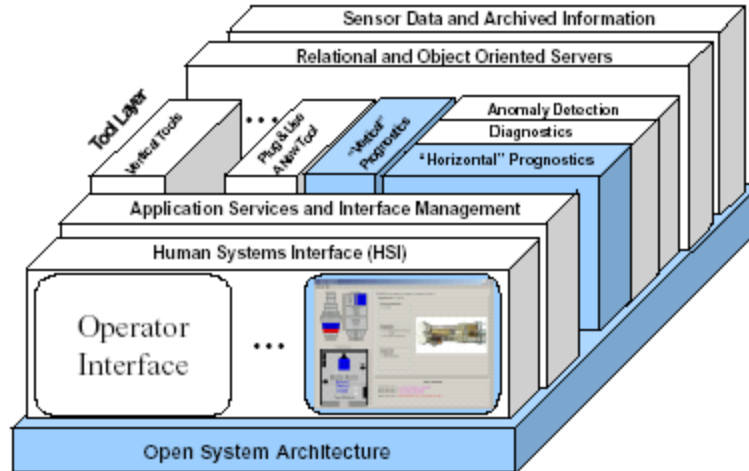


Figure 4: “Plug and Play” Toaster Model.

2.1.3 Open Questions

A flexible and extensible architecture for diagnostics and prognostics is not currently available to integrate results from different modules: sensor localization/selection, feature extraction, mode identification, diagnostics, and prognostics. Meanwhile, this architecture must specify a communication interface shared by different modules and a data sharing mechanism to exchange data between modules.

As a critical component in the architecture and another important issue in this research, the state-of-the-art of sensor localization/selection is reviewed in the next section.

2.2 Optimum Sensor Localization/Selection (SLS)

System performance is strongly dependent on available sensor measurements. Inaccurate measurements resulting from improper sensor localization/selection or insufficient

measurements can significantly deteriorate system performance. Therefore, SLS has received considerable attention and has been studied in different areas.

Traditionally, sensors are placed mainly to meet control or monitoring objectives. Faulds and Belinda [1] introduced the sensor placement problem for feedback control. Al-Shehabi and Newman [2] employed root locus principles to choose optimum sensor positions for aeroelastic vehicle feedback control applications. Giraud and Jouvencel [10] addressed the problem of sensor selection in an automatic task, such as a process of data fusion, a sensing task or the design of a perceptual system for a mobile robot. Chen and Li [14] presented an automatic sensor placement technique for robot vision in inspection tasks. One aspect of the NASA Aircraft Morphing program was to determine the optimum number of active control devices (for example, piezoelectric actuators) and their placement in the structure. In this program, Padula and Kincaid [3] provided a good review of sensor and actuator placement problems (Table 1). They grouped the sensor and actuator placement research based on different types of applications (non-aerospace placement problems and aerospace placement problems).

Table 1: Review of sensor placement methods [3].

First Author	Journal	Volume (Issue)	Year	Optimization Method	Potential Locations	Number Selected
Naimimoasses	<i>Measurement Science Tech.</i>	6(9)	1995	other	6	2
Oh	<i>Nuclear Eng. And Design</i>	152(1)	1994	other	52	14
Reis	<i>J. Water Resources Plan. & Manag.</i>	123(6)	1997	GA	25	3
Mattingly	<i>Int. J. Hyperthermia</i>	14(4)	1998	none	NA	4
Emery	<i>J. Heat Transfer</i>	119(4)	1997	none	NA	NA
Sunar	<i>AIAA Journal</i>	34(10)	1996	none	NA	2
Liang	<i>J. Intelligent Mat. Sys. & Structures</i>	6(4)	1995	none	8	3
Bhargava	<i>J. Intelligent Mat. Sys. & Structures</i>	6(3)	1995	none	5	2
Kang	<i>AIAA Journal</i>	34(9)	1996	none	NA	2
Hac	<i>J. Sound and Vibration</i>	167(2)	1993	none	10	2
Wang	<i>J. Acoustical Society of America</i>	90(5)	1991	none	NA	3
Kang	<i>AIAA Journal</i>	36(9)	1998	NLP	NA	1
Papadopoulos	<i>AIAA Journal</i>	36(2)	1998	other	NA	NA
Simpson	<i>Noise Control Engineering J.</i>	44(4)	1996	GA	40	15
Gawronski	<i>J. Sound and Vibration</i>	208(1)	1997	other	36	2
Kammer	<i>J. Sound and Vibration</i>	171(1)	1994	other	321	15
Pottie	<i>Internoise 96</i>		1996	GA	90	14
Manolas	<i>Internoise 96</i>		1996	GA	80	48
Wang	<i>J. Acoustical Society of America</i>	99(5)	1996	GA	NA	3
Katsikas	<i>Mechanical Sys. & Signal Processing</i>	9(6)	1995	GA	80	26
Maghami	<i>IEEE T. on Aero. & Elect. Sys.</i>	29(2)	1993	NLP	98	3
De Fonseca	<i>SPIE</i>	3041	1997	NLP	NA	4
Dhingra	<i>Int. J. for Num. Methods in Eng.</i>	38(20)	1995	GA	12	5
Wang	<i>J. Intelligent Mat. Sys. and Structures</i>	5(1)	1994	NLP	NA	3
Liu	<i>J. Aerospace Eng.</i>	10(3)	1997	SA	90	4
Chattopadhyay	<i>SPIE</i>	3329	1998	GA	NA	NA
Kincaid	<i>Location Science</i>	1(2)	1993	SA	1608	32
Kincaid	<i>J. Combinatorial Optimization</i>	1(3)	1997	TS	102	32
Furuya	<i>J. of Spacecraft and Rockets</i>	33(3)	1996	GA	1608	8

Some recent works have been concerned with other nontraditional activities, such as target tracking, fault detection, and reliability analysis. This research mainly focuses on sensor placement for fault diagnosis.

Sensor localization/selection for fault diagnosis has been studied at two different levels: component level vs. system level (Figure 5). Some of the sensor placement problems attempted to position sensors in a component's range [1][5], for example, a bearing or an object in 3D view. Critical systems of interest are characterized as large-scale systems consisting of multiple components. For such systems, a fault may propagate through several components when it occurs. Therefore, it is possible that sensors can be placed at any of the components to detect the fault. With hundreds or thousands of possible locations of sensors in a system, the selection of a crucial and optimum sensor location, sensor types, and number of sensors poses an important problem that needs to be solved at the system level before the detailed spatial distribution in a component can be determined.

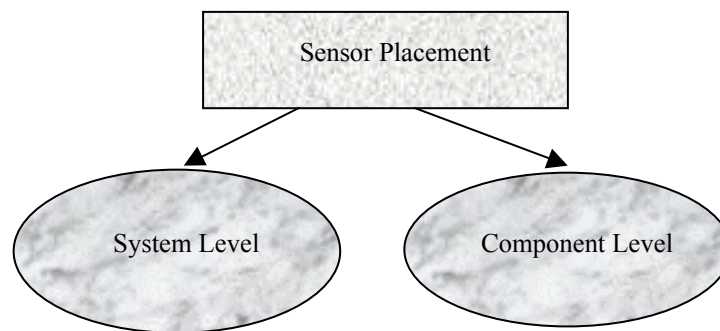


Figure 5: Different views of Sensor placement.

Although a large body of research work has emerged, the various approaches vary only in their choices of the three basic components (Figure 6): model, Figure-of-Merit, and optimization algorithm. The following discussion of sensor placement is based on these three components.

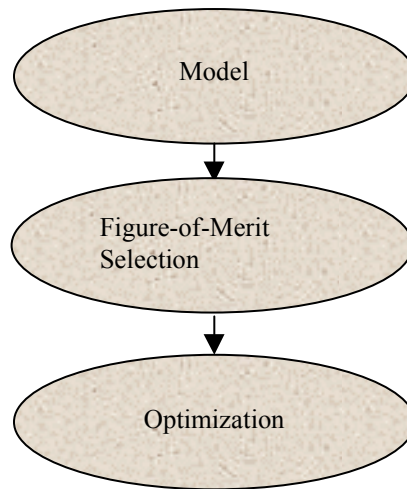


Figure 6: Basic components of sensor placement for fault diagnosis.

2.2.1 Component Level Sensor Localization

Models used at the component level usually are mathematical models or data-driven models. In a mathematical model, the physical system is described through the application of scientific principles. For example, Faulds and King [1] used a partial differential equation formulation as their system model. A mathematical model is precise without considering disturbances and is suitable for qualitative and quantitative analysis. However, building a mathematical model requires a thorough understanding of the

physical system, a difficult task for a complex system with highly nonlinear dynamics coupling a variety of physical phenomena in the temporal and spatial domains. A data-driven model is a black-box model that requires a large number of training data. Many intelligent tools can be used as the modeling tools. For example, a neural network is trained when it is fed enough template data and makes a decision based on the knowledge it has learned from the data. Theoretically, given enough training data, the neural network should be able to simulate a real system very well. As an example, a Multi-Layer Perceptron (MLP) is employed in Naimimohasses's work to model a system [5].

An objective function or a Figure-of-Merit (FOM) needs to be defined. Lim [30] determined the sensor locations for the purpose of disturbance rejection. Wang and his coworker [4] adopted the Signal-to-Noise Ratio (SNR) as the FOM in their study of a sensor placement strategy for in-situ bearing defect detection. Sensitivity of the system was used as a FOM in [5].

After a FOM is selected, an algorithm needs to be decided to optimize this FOM. Various optimization algorithms, from random search to heuristic algorithms such as Genetic Algorithms (GAs), have been used for optimizing the sensor location. Random search is suitable for a small and simple sensor placement problem since it is straightforward and easily implemented. But it is time consuming and inefficient when dealing with a large system. A Simulated Annealing (SA) method [3][20] based on random search is used to select a single random subset while seeking to improve the cost function by moving to one of the nearest neighbors of the selected subset. The Tabu search method [21] uses probabilistic events for the search of better solutions. GAs, based on the Darwinian principle of natural selection, are widely applied in different

domains, among the heuristic methods. Sen [31] designed his sensor network of the linear mass flow process using the GA method. Other examples using the GA method to solve sensor placement problems can be found in [6] and [7].

2.2.2 System Level Sensor Localization/Selection

More recently, interest has been focused on the sensor placement problem for fault diagnosis at the system level.

For the purpose of Fault Detection and Identification (FDI), Xu and Jiang [8] addressed a systematic analysis on where to pick-up the best signal to generate the residual: the controller outputs, the plant outputs, or some other locations (Figure 7). The control system is modeled by transfer functions. Recently, Cause-Effect (CE) analysis methods, such as graph theory, Petri-Net method, and fault tree method, have been widely used as the modeling tools for the sensor placement problem for fault diagnosis at the system level because of their simple graphical representations of the process. Raghuraj [9] used a Directed Graph (DG) model to represent the CE behavior of the process. Wang, Song, and Li built a DG model to represent the cause and effect relationships of process variables [19]. Another popular model is the Signed Directed Graph (SDG). Its structure is identical to that of DG and signs are placed on the arcs of a DG to obtain an SDG [23].

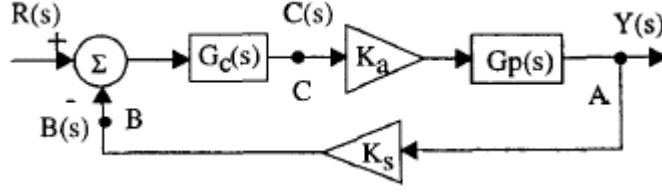


Figure 7: Closed-loop control system.

Many factors can contribute to a FOM. Xu and Jiang [8] adopted a sensitivity approach to build the FOM. The optimal location to obtain information for detecting different system component faults was determined based on the criteria of maximal sensitivity with respect to the variations of that component by analyzing Bode's sensitivity function. Raghuraj [9] and Wang [19] used fault observability and fault resolution to build the FOM. Observability refers to the condition that every fault defined for the process can be observed by at least one sensor, and fault resolution is defined as the ability to identify the exact fault that has occurred. In Raghuraj's work, the optimization objective is to maximize fault observability and resolution. As an extension to Raghuraj' work, Bhushan and his coworkers [23] used observability and reliability as the FOM for locating sensors. A fault monitoring system is considered highly reliable if the probability of any fault occurring without being detected is low. With the philosophy that a chain can be no stronger than its weakest link, Bhushan *et al.* proposed to maximize the minimum reliability for fault detection among all the faults. Because the faults of the process have certain probabilities of occurrence and the various available sensors have certain probabilities of failure, a fault can occur without being detected if all the sensors detecting that fault fail simultaneously. The reliability of detecting a fault is

inversely proportional to the unobservability value of that fault, which is given as the product of the fault occurrence and corresponding sensor failure probabilities:

$$U_i = f_i \prod_{j=1}^n (s_j)^{d_{ij} x_j} \quad (2-1)$$

where

f_i : the probability of the occurrence of fault i ;

s_j : the probability of the failure of sensor j ;

x_j : the number of sensors to be placed on node j ;

d_{ij} is non-zero if fault i affects node j .

Bagajewicz [22] considered cost as the FOM. He presented a minimum cost model for the design of reliable sensor networks:

$$\begin{aligned} & \text{Min} \sum_{i \in M_1} c_i q_i \\ & \text{s.t.} \\ & R_k(q) \geq R_k^* \quad \forall k \in M_R \\ & E_k(q) \geq E_k^* \quad \forall k \in M_E \\ & q_i = (0,1) \quad \forall i \end{aligned} \quad (2-2)$$

Reliability of specific variables, as well as other constraints, such as bounds on the degree of observability and precision of key variables, can be easily incorporated into the minimum cost model [24][25][26].

In addition to the optimization approaches mentioned in Section 2.2.1, a graph-based method is utilized as the optimization tool for sensor placement for fault diagnosis at the system level [9]. A greedy search algorithm based on a bipartite graph was proposed by Raghuraj [9] to generate a minimal set of sensors to meet observability and maximum resolution requirements. A bipartite graph is one whose vertex set can be partitioned into

two sets in such a way that each edge joins a vertex of the first set to a vertex of the second set. Whenever there is a directed path from a root node to a key component, an arc from that root node to the key component is drawn in the bipartite graph. Two optimization algorithms based on graph theory are listed in the Appendix A and Appendix B.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [64], inspired by social behavior of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. The fitness value is also stored. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighborhood of the particle. This location is called "lbest." When a particle takes all the population members as its topological neighbors, the best value is a global best and is called "gbest" [65].

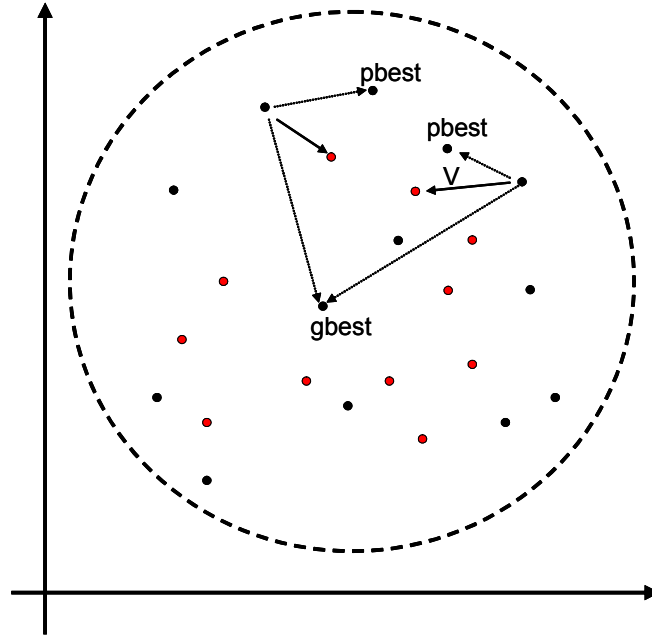


Figure 8: Particle swarm optimization.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO), as shown in Figure 8. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations.

In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that are employed across a wide range of applications, as well as for specific applications focused on a specific requirement.

2.2.3 Sensor Uncertainty

At the fault initiation stage, the symptoms are usually difficult to be detected due to uncertainty. Therefore, uncertainty analysis is a key factor for selecting sensors for fault diagnostic purposes and should be used to assist in the selection of sensors based on their relative performance.

Sources of Error

Uncertainty arises from three major sources:

- Environment uncertainty [67] is a common type of source of uncertainty. Air temperature, wind, and barometric pressure are examples of the sources of environment uncertainty.
- Model uncertainty is produced due to the model inaccuracy. The selection of modeling techniques may also lead to uncertainty in a system.
- Measurement uncertainty, the final source of uncertainty, is introduced during the measurement process: human perception, calibration error, sensor error, etc.

This research will focus on the measurement uncertainty related to sensors. Environment uncertainty may impact the measurement results through noise and interference. This type of uncertainty will be considered as a part of measurement uncertainty as well.

The sensor measurement process consists of three distinct steps: calibration, data acquisition, and data reduction. Errors that enter during each of these steps will be grouped under their respective error source heading: calibration errors, data acquisition errors, and data reduction errors.

Common elemental calibration errors include:

- Primary to interlab standard
- Interlab to transfer standard
- Transfer to lab standard
- Lab standard to measurement system
- Calibration technique, etc.

Data acquisition error sources:

- Measurement system operating conditions
- Sensor-transducer stage (instrument error)
- Signal conditioning stage (instrument error)
- Output stage (instrument error)
- Process operating conditions
- Sensor installation effects
- Environment effects
- Spatial variation error
- Temporal variation error

Data Reduction Error:

- Calibration curve fit
- Truncation error

Representation of Uncertainty

All measurements of a variable contain inaccuracy or uncertainty. We will take the word accuracy to refer to the closeness of agreement between a measured value and the true value. The degree of inaccuracy or the total measurement error (δ) is the difference between the measured value and the true value. The total error is the sum of the

systematic (or bias) error and the random (or precision) error. The systematic error (β) is the fixed or constant component of the total error and is sometimes referred to simply as the bias. The random component (ε) of the total error is sometimes called the repeatability, repeatability error, or precision error. Therefore, uncertainty can be grouped into two general categories: bias error and precision error. Measurement blunders that result in obviously fallacious data will not be considered.

The precision error, or random error, of a measurement system refers to the ability of the system to indicate a particular value upon repeated but independent applications of a specific input value. Precision error is a measure of the random variation to be expected during such repeatability trials.

Random error is based on the assumption that the distribution is Gaussian or normal:

$$p(x) = \frac{1}{\sigma(2\pi)^{1/2}} \exp\left[-\frac{1}{2} \frac{(x - x')^2}{\sigma^2}\right] \quad (2-3)$$

$$\sigma = \lim it \left\{ \left[\sum (X_i - \mu)^2 / n \right]^{1/2} \right\} \quad (2-4)$$

However, an experimenter never has at his disposal an infinite population of data, but just a sample of n data points with which to calculate the standard deviation:

$$S_x = \pm \left\{ \left[\sum (X_i - \bar{X})^2 / (n-1) \right]^{1/2} \right\} \quad (2-5)$$

Where:

X_i : the value of the i th X in the sample

\bar{X} : the sample average

(n-1): the degrees of freedom for the sample.

The interval $(\bar{X} \pm S_x)$ contains $\approx 68\%$ of the data sample. When there are less than 30 degrees of freedom, S_x cannot be used. Instead, “t” distribution is used. Usually, $(\bar{X} \pm t_{95} S_x)$ contains 95% of the data sample.

The average error in a series of repeated calibration measurements defines the error measure known as a bias. Bias error is the difference between the average and true values. The bias error, β , is never known, its limits is estimated with the bias limit, B , in which

$$-B \leq \beta \leq B \quad (2-6)$$

Measurement uncertainty is defined as the combination of both the bias and precision errors of uncertainty. Two methods are provided for that combination: the addition model and the root-sum-square model.

The Addition (Add) Uncertainty Model ((Symmetrical Bias Limits))

$$U_{ADD} = \pm[B + t_{95}S] \quad (2-7)$$

The Root-Sum-Square (RSS) Uncertainty Model (Symmetrical Bias Limits)

$$U_{RSS} = \pm[B^2 + (t_{95}S)^2]^{1/2} \quad (2-8)$$

From statistical analysis of the data set and an analysis of sources of error that influence this data, we can estimate x' as

$$x' = \bar{x} \pm \mu_x \quad (2-9)$$

Where \bar{x} represents the most probable estimate of x' based on the available data and μ_x the confidence interval or uncertainty in that estimate at some probability level (P %), usually 95%. based both on an estimate of the precision error and the bias error in the measurement of x . Either U_{ADD} or U_{RSS} can be used as μ_x .

Two stages are of concern while considering the effect of sensor uncertainty on the sensor fault detectability: design stage and implementation stage.

Design Stage Uncertainty Calculation

In every design there is an initial stage where the measurement system and its procedure are but a concept. At this point in the design, what is usually known about the sensors stems from the manufacturers and previous experience.

In the design stage, distinguishing between bias and precision errors is too difficult to be of concern. Instead, consider only sources of uncertainty in general. Even when all errors are otherwise zero, the value of the measurand is affected by the ability to resolve the information provided by the instrument. This is called the zero-order uncertainty of the instrument, μ_0 . At the zeroth order, it is assumed that the variation expected in the measurand will be less than that due to instrument resolution and that all other aspects of the measurement are perfectly controlled. Basically, μ_0 is an estimate of the expected uncertainty caused by the data scatter due to reading the instrument.

As an arbitrary rule, assign a numerical value to μ_0 of one-half of the instrument resolution with a probability of 95%:

$$\mu_0 = \frac{1}{2} \text{ resolution} \quad (2-10)$$

The second piece of information that is usually available is the manufacturer's statement concerning instrument errors. The value stated in specification is some typical value for a specific instrument under ideal conditions. We can assign this stated value as the uncertainty due to the instrument, μ_c . Sometimes the instrument errors will be stated in parts, each part due to some contributing factor. A probable estimate of μ_c can be made by treating the propagation of errors to the instrument uncertainty using the error propagation law. This is an estimate of the bias error due to the instrument.

The design-stage uncertainty, μ_d , for the instrument can be approximated by combining the instrument uncertainty with the zero-order uncertainty:

$$\mu_d = \sqrt{\mu_0^2 + \mu_c^2} \quad (\text{P}\%) \quad (2-11)$$

Implementation-Stage Uncertainty Calculation

In design-stage uncertainty analysis, only errors due to a measurement system's resolution and estimated intrinsic errors are considered. Implementation-stage uncertainty analysis takes one step further by considering procedural and test control errors that affect the measurement.

For zero-order uncertainty, all variables and parameters that affect the outcome of the measurement, including time, are assumed to be fixed except for the physical act of observation itself. Thus:

$$\mu_0 = \frac{1}{2} \text{ resolution} \quad (2-12)$$

Higher-order uncertainty estimates consider the controllability of the test operating conditions. For example, at the first-order level, the effect of time as an extraneous variable in the measurement might be considered:

$$\mu_1 = \pm t_{v95} S_X \quad (2-13)$$

As the final estimate, instrument calibration characteristics are entered into the scheme through the instrument uncertainty, μ_c , given by

$$\mu_N = \sqrt{(\mu_c)^2 + \sum_{i=1}^N \mu_i^2} \quad (95\%) \quad (2-14)$$

Uncertainties Analysis Techniques

The use of an analysis approach to estimate the effect of uncertainties is referred as uncertainty propagation. Several categories of methods exist in the literature. The first category is the conventional sample-based approach such as Monte Carlo Simulations (MCS) [68]. Monte Carlo simulation provides an efficient approach for integrating and propagating probability distributions through a risk model. Although alternative sampling techniques such as Quasi Monte Carlo Simulations including Halton sequence, Hammersley sequence, and Latin Supercube Sampling have been proposed, none of these techniques are computationally feasible for problems that require complex computer simulations, each taking at least a few minutes or even hours or days [69]. The second category of uncertainty propagation approach is based on analytical analysis [70]. Some of the widely used analytical methods for sensitivity/uncertainty are: (a) differential analysis methods, (b) Green's function method, (c) spectral based stochastic

finite element method, and (d) coupled and decoupled direct methods. The third category of uncertainty propagation approach is computer algebra based methods. Computer algebra based methods involve the direct manipulation of the computer code [70], typically available in the form of a high level language code (such as C or FORTRAN), and estimation of the sensitivity and uncertainty of model outputs with respect to model inputs. These methods do not require information about the model structure or the model equations, and use mechanical, pattern-matching algorithms, to generate a "derivative code" based on the model code. One of the main computer algebra based methods is the automatic differentiation, which is sometimes also termed automated differentiation.

2.2.3 Open Questions

Previous reported work in this area does not address some key concerns:

- To determine the sensor localization for fault diagnostic purposes, the severity of failure effects and the probability of occurrence of failures must be considered.
- Sensor uncertainty must be accounted in the research of SLS for fault diagnostic purposes.
- Qualitative models, such as DG and SDG, are unable to express some important non-qualitative information: how large the SNR at a sensor node is; how sensitive and how fast a sensor node can measure a fault.
- Most of the available research findings are aimed at designing sensor locations in a new system, assuming there is no sensor installed in the current system. It is not always true in a current industrial environment. Many systems have sensors on board already, though they may be used for other purposes, such as measurement and control. In these situations, the objective of sensor placement for fault diagnosis is to

select available sensors as well as to design additional sensors needed for FDI purposes.

2.3 Research Focus

The research focuses on the following topics:

- An integrated architecture for diagnostics and prognostics is proposed and implemented.
- Fault detectability is defined to quantify the sensor's ability for fault detection, considering sensor uncertainty.
- A Quantified Directed Graph (QDG) model is proposed to model the fault propagation in a system.
- A generalized Figure-of-Merit is defined for the sensor localization/selection problem for fault diagnosis at the system level and a sensor localization/selection method will be introduced based on particle swarm optimization method.
- Validation of the proposed sensor localization/selection approach in the integrated diagnostic/prognostic architecture.

CHAPTER 3

INTEGRATED DIAGNOSTIC/PROGNOSTIC ARCHITECTURE

Diagnostics and prognostics are complex tasks that involve various functional components, such as sensor localization, sensor selection, data acquisition, feature extraction, diagnosis, and prognosis. As such, it is important to design an open and flexible architecture. With contributions from my colleagues [63], a proposed integrated system architecture for diagnostics and prognostics is shown in Figure 9.

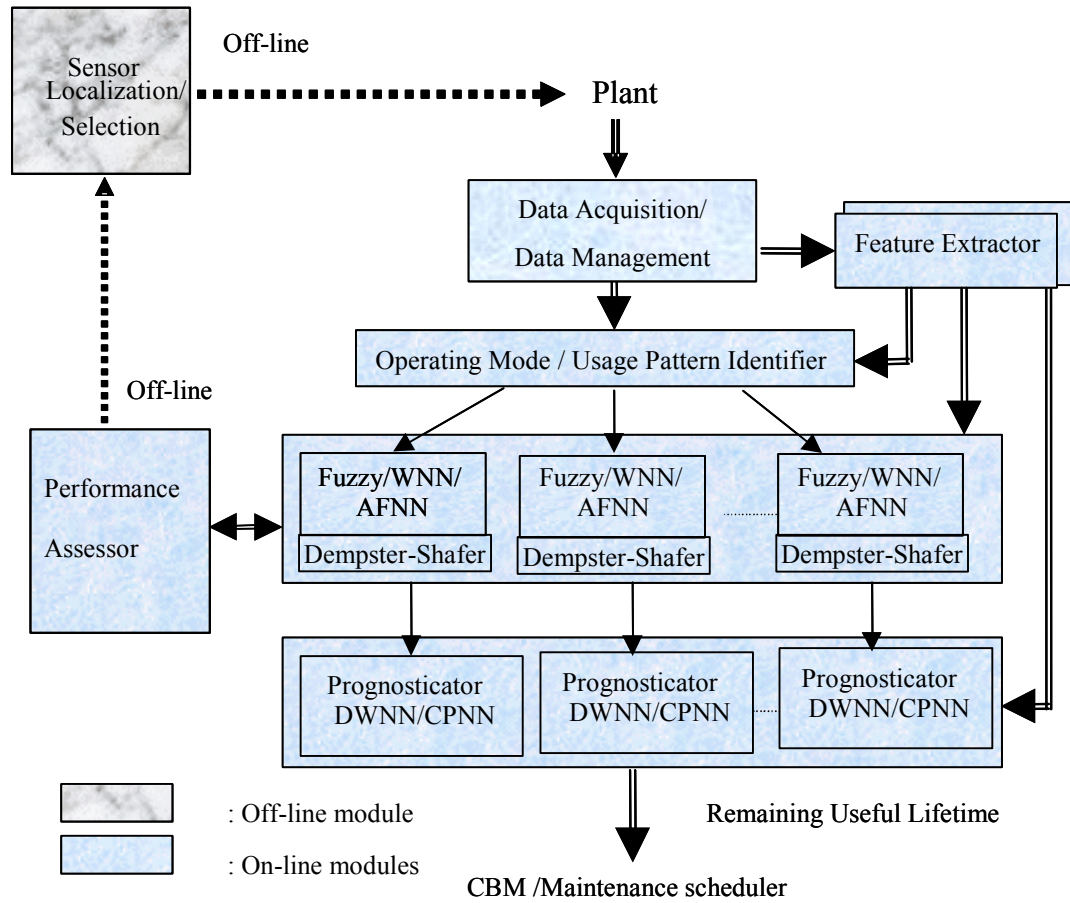


Figure 9: Integrated system architecture for diagnostics/prognostics.

My contributions to the architecture include the following:

- Design of the sensor localization/selection module.
- System Integration. Most of the modules in the architecture have been developed independently and their data structures and interfaces are not consistent. The integration at the system level is aimed to manage all the separate modules.
- Database management & user interface development. A database management module is developed to efficiently store and access sensor data, features, and

diagnostic and prognostic results and a graphic user interface is developed to provide real-time interface to display, configuration, and operate the system.

3.1 Subsystem Modules

As shown in Figure 9, this architecture consists of several on-line modules and an off-line module. In the on-line phase, complete diagnostics and prognostics are implemented using various functional components as follows:

- Data Acquisition (DAQ)/Data Management Module: provides an interface for data acquisition from real systems or other software.
- Feature Extractor (FE) [61]: extracts useful information in the form of a feature vector from the raw sensor data according to the requirements imposed by diagnostic and prognostic modules.
- Operating Mode / Usage Pattern Identifier [62]: decides upon a specific operating mode and usage pattern of the system.
- Classifier: employs the Dempster-Shafer theory as a knowledge fusion tool to combine the classification results from a fuzzy inference engine, static Wavelet Neural Networks (WNN [59]), and an Arrangement Fuzzy Neural Network (AFNN) [61]. The occurrence of a fault mode is determined and identified on-line based upon the fusion result.
- Prognosticator: capitalizes upon a virtual sensor to provide fault dimensions and a dual approach to prediction — a Dynamic Wavelet Neural Networks (DWNN [59]) (supervised/unsupervised) for fault trending and a Confidence Prediction Neural Network (CPNN [58]) assisted by the Fuzzy Analytic Hierarchy Process (FAHP)

aimed primarily at accommodating causal adjustments to the prediction curve and managing uncertainty bounds.

- Performance Assessor: assesses the performance of the sensor localization/sensor selection/diagnostic/prognostic modules.

One important aspect of the architecture is to divide a dynamic system into subsystems according to the operating modes/usage patterns. Each subsystem has a classifier and a prognosticator with a limited number of inputs, thereby reducing not only the number of neurons in the network, but also the run time and training time required for the network, contributing to improved speed for failure isolation and detection. System resource consumption is also reduced as well as computational complexity. Moreover, instead of using one single classifier for the whole system, fewer changes to the existing networks are required when additional classifiers are added to the central database as new features are derived, new devices are inserted, or new operating modes/usage patterns are defined. In this way, the architecture is open and flexible. Finally, if a hybrid system can be divided into several continuous sub-systems according to operating modes or usage patterns, the diagnostic and prognostic issues in the hybrid system can be addressed by dealing with the problems in subsystems in the continuous domain.

The sensor localization/selection (SLS) module is included at the off-line phase. The SLS module determines optimum types, location, and number of sensors for fault diagnosis at the system level.

3.2 Software Architecture

The software architecture is shown in Figure 10. All the modules introduced in Section 3.1 are built into a Component Object Model (COM)/Distributed Component Object

Model (DCOM) infrastructure in order to be accessed in an open and flexible manner accommodating a variety of development languages.

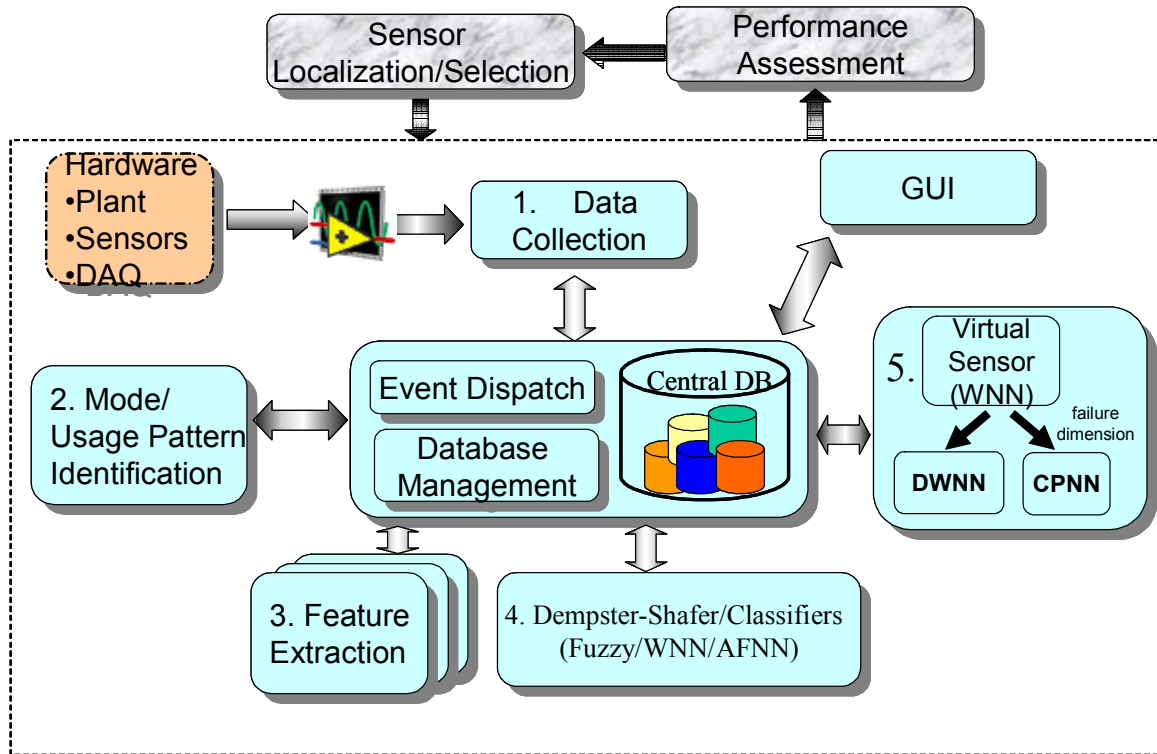


Figure 10: Software architecture.

This software architecture has several characteristics:

- The architecture is scalable and language independent.
- The architecture is open and flexible. Components can be developed and upgraded independently. This component-based architecture encourages flexible “plug-and-play” extensibility and evolution of systems. New components can be added easily when new algorithms are developed.

- Real time distributed computing can be realized.
- Various products and technologies (such as Integrated Condition Assessment System, or ICAS, prognostic modules, and other plug-and-play modules) can work together with maximum compatibility and efficiency.
- Event driven and call back technologies increase the communication efficiency among components.

3.3 COM/DCOM

The modules, including feature extractor, operating mode / usage pattern identifier, classifiers, and prognosticators in this diagnostic/prognostic architecture are developed in different programming languages, such as Matlab, C/C++, and Visual Basic. It is inefficient and not necessary to rewrite all the modules using a unified programming language. This raises a problem in integrating all the modules. Among others, the Component Object Model (COM)/Distributed Component Object Model (DCOM) structures defined by Microsoft Corporation are able to solve this problem.

The COM supports interoperability and reusability of distributed objects by allowing developers to integrate objects from different vendors in the same machine. DCOM is an extension of COM that allows network-based component interaction. Under the COM/DCOM structure, all the current modules in different programming languages can be utilized if they are transferred into the COM/DCOM structure. Also, for those future modules, no strict programming language requirements are enforced.

3.4 Database Management

A relational database management module is developed as a COM object to efficiently store and access raw data, features, diagnostic and prognostic results, and system

configuration parameters. The database interface provides functions to access the central database by means of a COM interface and to simplify the database programming and management. It hierarchically relates the main system to its components, which in turn relates to the sensors and their data and fault conditions. This scheme provides high cohesiveness, while at the same time isolating the actual database from the rest of the system.

The speed and performance of the database interface are crucial requirements since every component of the diagnostic/prognostic system is accessing the central database. Several improvements have been made since its initial design, resulting in significant efficiency improvements.

3.5 Graphical User Interface (GUI) and Communication

A user-friendly GUI displays real-time values and historical records of sensors and features, as well as diagnostic/prognostic results. Meanwhile, the diagnostic/prognostic functions can be accessed through this interface. The communication between modules is event based and does not require an overall scheduler to manage all the modules.

- The architecture's main routine runs continuously at a nominal rate (typically same as the real sampling rate of the data acquisition system). In each cycle, data are collected from external data sources to the central database through API, drivers, or database queries. Also, features are extracted continuously.
- When the features are ready for all the fault modes of interest, a "feature ready" event is fired.
- With the firing of a "feature ready" event, diagnostic modules begin to determine whether a fault mode exists or not. If a fault is detected, the "fault detected" event is

fired after the fault is confirmed. Otherwise, if no fault is detected, all the modules are in an idle state until the next cycle.

- With a “fault detected” event, the prognostic modules are activated to predict the time-to-failure and related information and an event of “prognostics ready” is fired.

CHAPTER 4

SENSOR LOCALIZATION/SELECTION FOR FAULT DIAGNOSIS

The purpose of sensor localization/selection (SLS) in this research is to specify the types, location, and number of sensors for fault diagnosis. SLS for fault diagnosis can be studied from two perspectives: system level and component level, as shown in Figure 11. Since component-level SLS relies greatly on the sensor types and is usually application specific, only system-level sensor localization/selection is studied in this research.

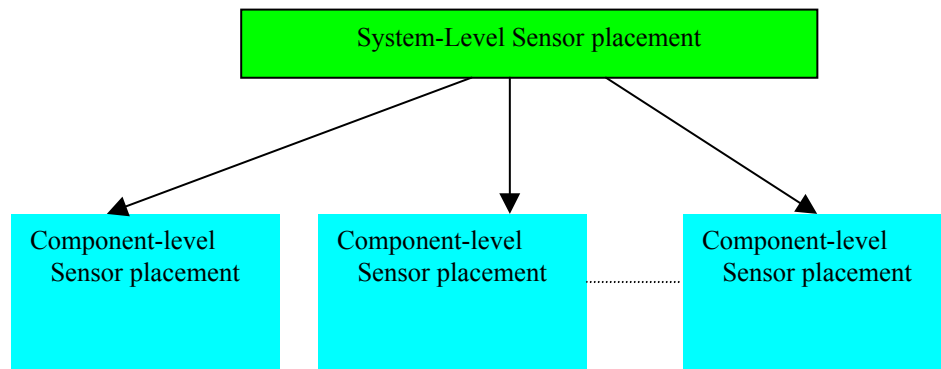


Figure 11: Sensor localization/selection hierarchical structure.

4.1 Sensor Localization/Selection Architecture

In this research, sensor localization/selection entails several functional modules: requirements analysis, Failure Mode and Effects Criticality Analysis (FMECA) study,

quantitative model, Figure-of-Merit (FOM), optimization, and performance assessment, as shown in Figure 12. This figure illustrates how the sensor localization/selection modules are integrated together and interconnect with a diagnostic/prognostic system.

First, in the requirement analysis module, sensor localization/selection requirements are analyzed and basic definitions needed for other modules are provided.

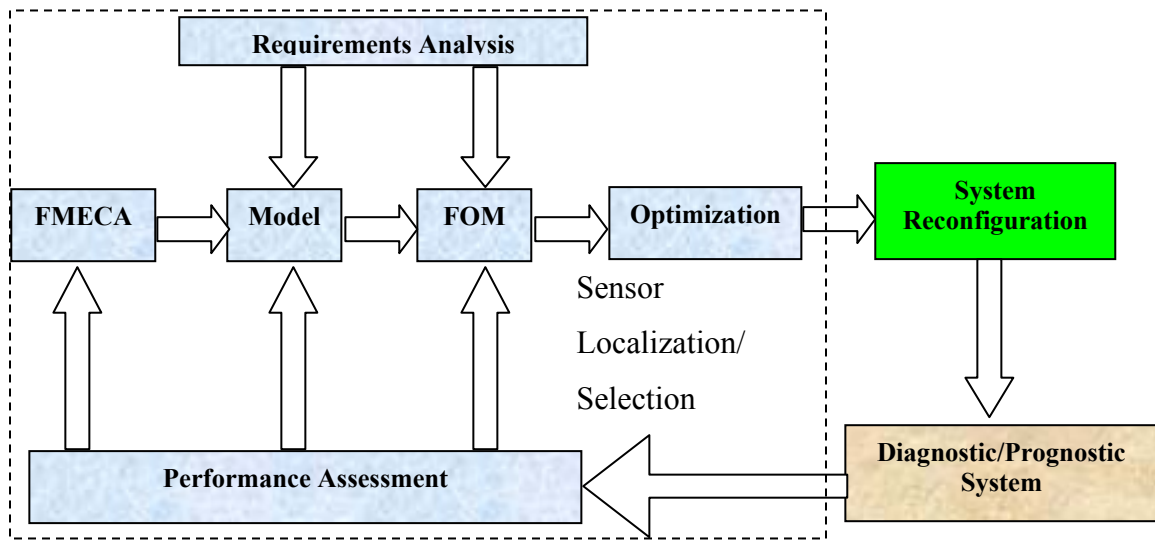


Figure 12: Sensor localization/selection architecture.

Secondly, since the purpose of sensor localization/selection in our research is to achieve maximum fault detection performance with applicable constraints, it is critical to analyze and understand each fault mode/failure mode. A FMECA study is widely employed to identify failure modes and specifies severity of failures and frequency of their occurrence.

Current research work on sensor localization/selection is focused on qualitative analysis that usually leads to a qualitative fault propagation model. With the information from the FMECA study and the definition given in the requirements analysis module, a quantitative model, Quantified-Directed-Graph (QDG) model is introduced to build the fault propagation model quantitatively in the proposed architecture.

The thorough analysis of the sensor localization/selection requirements is also employed to establish a generalized Figure-of-Merit (FOM) to cover the most important issues in current research. Having selected a generalized FOM, a popular evolution computation technique, particle swarm optimization, is employed considering the rich heuristic information during sensor localization/selection process.

Finally, with selected sensors for fault diagnosis, systems are configured for on-line diagnostics/prognostics and the diagnostic performance is evaluated through a performance assessment module. Meanwhile, the feedback information from the performance assessment module is utilized by the other sensor localization/selection modules to fine-tune the selected sensors and improve the on-line diagnostic performance.

In the following sections, these components in the architecture are discussed in detail.

4.2 Requirements Analysis

Four main requirements need to be met to optimally place sensors for fault diagnostic purposes: detectability, identifiability, fault detection reliability, and a requirement associated with limited resources. Meanwhile, sensor uncertainty needs to be addressed since uncertainty is always related to the sensor measurement capability.

4.2.1 Detectability

Definition: Total Fault Detectability (D) is defined as the extent to which the diagnostic scheme can detect the presence of a particular fault. The detectability of a fault relies on the sensors selected to detect the fault, therefore, “sensor fault detectability” is defined first.

Definition: Sensor Fault Detectability is the extent to which a sensor can detect the presence of a particular fault. It depends on the following factors:

Signal-to-Noise Ratio (SNR): A low SNR implies that it is hard for the sensor to detect the fault. Thus, the sensor fault detectability is poor by considering the SNR only. On the other hand, a high SNR implies a good sensor fault detectability if all other factors are equal.

Time-To-Detection to Time-To-Failure Ratio: “time-to-detection” is the time span between the initiation of a fault (potential failure) and the detection of the fault by the sensor; “time-to-failure” refers to the duration between the initiation of the fault and the time when the failure occurs. A low ratio implies that the sensor can detect the fault in a relatively short period of time and the sensor fault detectability is good; while a high ratio implies that a relatively long period of time is needed to detect the fault and the sensor fault detectability is poor. Also note that, if the ratio is equal to or greater than 1, i.e., the fault is detected by a sensor when or after the fault leads to a failure, for this case we would have no reason to select this sensor to detect the fault and the sensor fault detectability is regarded as 0.

Fault detection sensitivity of a sensor: if the sensor data can be used to map known measurements to a “fault measurement,” the fault detection sensitivity of the sensor can

be defined as the ratio of the change of the fault measurement to the change of the sensor measurement: $\frac{\Delta s}{\Delta f}$, where Δs is the change of the sensor measurement, and Δf is the change of the fault measurement. The sensor fault detectability increases with the fault detection sensitivity of the sensor.

The definition of fault detection sensitivity must account for the resolution of a sensor and the “resolution” of a fault, which is the minimum measurable fault of interest. For each fault, minimum observable symptoms can be determined based on the expert knowledge or past experiences. Thus, the definition of the fault detection sensitivity of a sensor can be reexpressed as:

$$\text{Sensitivity} = \frac{\Delta s / \text{Resolution}}{\Delta f / \Delta f_{\min}} \quad (4-1)$$

Let us now pose the following example for clarity. Let there be three sensor nodes A, B and C, in the presence of a fault f_1 . The fault changes 1 unit, which is also the minimum measurable fault, the sensor measurement changes A, B, and C, are 0.1, 0.2, and 0.4 units, respectively, a resolution of sensors A, B, and C are 0.01, 0.001, and 1, respectively.

Not considering the resolutions of these sensors, the fault detection sensitivity would be 0.1, 0.2, and 0.4. Sensor C is most sensitive to the fault. However, the change of sensor C is smaller than its resolution. Thus, it would not be able to pick up the true sensor value.

Taking into account both the sensor resolutions and the sensor measurement changes, we can update the calculation for this example: the fault detection sensitivity of sensors A, B, and C would be 10, 200, and 0.4, respectively.

Symptom Duration to Time-to-Failure Ratio: “symptom duration” is the time period of a fault affecting the measurement of a sensor. In most cases, if a fault occurs, affected sensors can measure the fault until the fault evolves to a failure. But it is possible that some sensors may return to their normal measurement states after a short time period. In this case, the symptom duration to time-to-failure ratio is small, and these sensors may not be good for detecting the fault.

Based on empirical experiences and these factors, the sensor fault detectability is normalized within 0 to 1 and given by the following equation:

$$SD_{ij} = \begin{cases} (1 + e^{-b*(V_{ij}-c)})^{-1} * (1 + e^{-g*(SNR_j-h)})^{-1} * (1 - \frac{TTD_{ij}}{TTF_{ij}})^\alpha * (\frac{SyD_{ij}}{TTF_{ij}})^\beta & \text{if } TTD_{ij} < TTF_{ij} \\ 0 & \text{if } TTD_{ij} \geq TTF_{ij} \end{cases} \quad (4-2)$$

Where,

i: Fault index; j: Sensor index;

V_{ij} : Fault detection sensitivity of sensor j for fault i;

b, c: Shape control parameters of fault detection sensitivity, application specific
with typical values of 10 and 0.5, respectively;

SNR_j : Signal-to-noise ratio of sensor j;

g, h : Shape control parameters of SNR, application specific with typical values of
1 and 0.5, respectively;

TTD_{ij} : Time-to-detection for fault i of sensor j;

TTF_{ij} : Time-to-failure for fault i of sensor j;

α : Shape control parameter of the time-to-detection to time-to-failure ratio,

application specific with a typical value of 0.5;

β : Shape control parameter of the symptom duration to time-to-failure ratio,

application specific, typical value is 0.2;

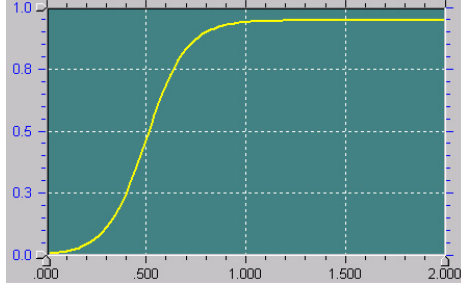
SyD_{ij}: Symptom duration for fault i of sensor j.

A typical sensor fault detectability is given by the following equation:

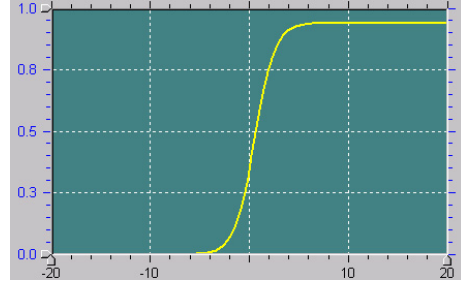
$$SD_{ij} = \begin{cases} (1 + e^{-10*(V_{ij}-0.5)})^{-1} * (1 + e^{-(SNR_j-0.5)})^{-1} * (1 - \frac{TTD_{ij}}{TTF_{ij}})^{0.5} * (\frac{SyD_{ij}}{TTF_{ij}})^{0.2} & \text{if } TTD_{ij} < TTF_{ij} \\ 0 & \text{if } TTD_{ij} \geq TTF_{ij} \end{cases}$$

(4-3)

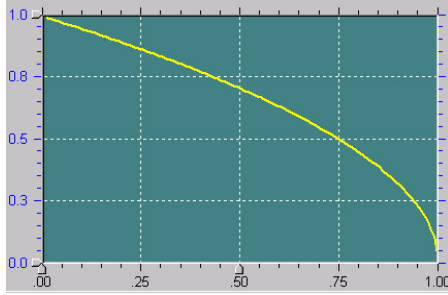
By varying the sensitivity, SNR, time-to-detection to time-to-failure ratio, and the symptom duration to time-to-failure ratio, different sensor fault detectability outcomes are given in Figure 13:



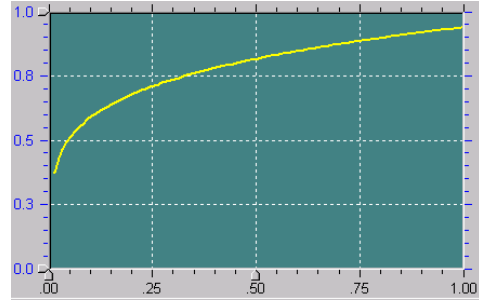
(A): Sensitivity-detectability.
(SNR = 20dB; TTD/TTF=0.1; SyD/TTF=1).



(B): SNR-detectability.
(V=1; TTD/TTF=0.1; SyD/TTF=1).



(C): TTD to TTF ratio vs. detectability.
(V = 1; SNR = 20dB; SyD/TTF=1).



(D): SyD to TTF ratio vs. detectability.
(V = 1; SNR = 20dB; TTD/TTF=0.1).

Figure 13: The relationships of detectability and the affecting factors.

From the plots of Figure 13, the sensor fault detectability increases with the SNR, the fault detection sensitivity, and the symptom duration, and decreases with the time-to-detection to time-to-failure ratio.

Based on the definition of sensor fault detectability, the total fault detectability of a fault can be computed.

If sensors in the set $A_i = \{S_{i1}, S_{i2}, \dots, S_{im}\}$ are selected to detect fault i , the detectability of fault i is defined as the mean value of the sensor fault detectability of all the sensors in A_i :

$$D_i = \frac{\sum_{j=1}^m SD_{ij}}{m} \quad (4-4)$$

4.2.2 Uncertainty Issues in Detectability

This thesis focuses on the measurement uncertainty which is related to sensors. Environment uncertainty may impact the measurement results through noise and interference. Thus, this type of uncertainty will be considered as a measurement uncertainty as well.

As in Equation 4-4, total fault detectability is defined as the mean value of the sensor fault detectability of those sensors selected to detect the fault. Therefore, the uncertainty of total fault detectability originates from the individual sensor fault detectability. Considering the four factors contributing to the sensor fault detectability, uncertainty plays an important role in the definition of fault detection sensitivity of a sensor and the SNR of a sensor. The others can be calculated statistically by averaging the experimental data.

The uncertainty affects the detection sensitivity. Also, only one type of error, random error, needs to be considered. This is because we are only interested in the change of sensor measurements, and the bias error would be removed from the change of measurements.

$$E\left(\frac{\Delta s}{\Delta f}\right) = E\left(\frac{\Delta(\bar{s} + \mu)}{\Delta f}\right) = \frac{\Delta E(\bar{s} + \mu)}{\Delta f} = \frac{\Delta(E(\bar{s}))}{\Delta f} + \frac{\Delta E(\mu)}{\Delta f} \quad (4-5)$$

The most common type of noise is the zero mean Gaussian white noise; therefore, the latter part of the above equation becomes 0, and:

$$E\left(\frac{\Delta s}{\Delta f}\right) = \frac{\Delta(E(\bar{s}))}{\Delta f} \quad (4-6)$$

For other types of noises, we can consider their effects using this SNR formulation.

The uncertainty of sensor fault detectability can be calculated by the law of propagation of uncertainty.

Assume Y is determined from N other quantities X_1, X_2, \dots, X_N through a functional relation f:

$$Y = f(X_1, X_2, \dots, X_N) \quad (4-7)$$

The output estimate y, which is the result of the measurement, is given by

$$y = f(x_1, x_2, \dots, x_N) \quad (4-8)$$

The combined standard uncertainty of the measurement result y, designated by $u_c(y)$ and taken to represent the estimated standard deviation of the result, is the positive square root of the estimated variance $u_c^2(y)$ obtained from the law of propagation of uncertainty:

$$u_c^2(y) = \sum_{i=1}^N \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) \quad (4-9)$$

Assume all the factors contributing to sensor fault detectability are independent to each other: sensitivity, SNR, time-to-detection to time-to-failure, and symptom duration to time to failure ratio, then only the second item in equation 4-9 is removed. Then the

uncertainty in the sensor fault detectability can be characterized by a random variable and the propagation law is applied:

$$u_c^2(SD) = \sum_{i=1}^4 \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) = SD^2 \times \left\{ b^2 \times (1 + e^{-b(x_1-c)})^{-2} \times e^{-2b(x_1-c)} \times u^2(x_1) + \right. \\ \left. g^2 \times (1 + e^{-g(x_2-h)})^{-2} \times e^{-2g(x_2-h)} \times u^2(x_2) + \right. \\ \left. \alpha^2 \times (1 - x_3)^{-2} \times u^2(x_3) + \beta^2 \times (x_4)^{-2} \times u^2(x_4) \right\} \quad (4-10)$$

The uncertainty of total fault detectability can be easily calculated as:

$$u_c(D_i) = \frac{\sum_{j=1}^m b_{ij} \times u_c(SD_{ij})}{\sum_{j=1}^m b_{ij}} \quad (4-11)$$

4.2.3 Identifiability

Since more than one fault may occur in a system, sensors should be placed not only to detect faults, but also to isolate or identify them. This assumption leads to the need for defining an identifiability term.

Claim 4.1: Assume fault i affects sensors in the set $A_i = \{S_{i1}, S_{i2}, \dots, S_{im}\}$, and fault k affects sensors in the set $A_k = \{S_{k1}, S_{k2}, \dots, S_{kq}\}$. If $A_i \cap A_k$ is empty, then, clearly, fault i and fault k can be easily distinguished. If $A_i \cap A_k$ is not empty, a virtual fault g is created that affects sensors in A_i or A_k , but not in both, i.e. in the set $B_g = A_i \cup A_k - A_i \cap A_k$. Under the single-fault assumption, if the faults i , k , and virtual fault g can be detected, then fault i and fault k can be identified.

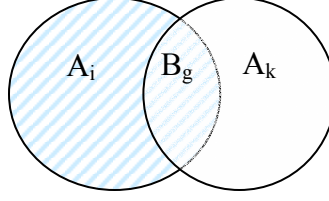


Figure 14: identifiability.

Proof of claim 4.1: if fault g can be detected by the sensors selected, i.e., the detectability of the virtual fault g is positive, then at least one sensor from set B_g is selected. This sensor is either in A_i or A_k , but not in both, which can imply that this sensor can measure either fault i or fault k , but not both, if a fault occurs. Thus, fault i and fault k can be identified.

This idea is also applicable to double-fault and multi-fault scenarios. For example, if two faults i and k can possibly occur at the same time, virtual faults measured by $A_{ik} = A_i \cup A_k$ can be generated and added to the FOM to solve the double-fault scenario.

The above claim is enough for fault identification without considering the sequence of fault symptoms. However, false positive rate may exist. Consider the following example.

Example 4-1: If fault f_1 occurs, sensor value S_1 decreases after time t_1 and S_2 increases after time t_2 ;

If fault f_2 occurs, sensor value S_1 decreases after time t_4 and S_3 increases after time t_3 .

To detect these two faults, f_1 and f_2 , with a minimum number of sensors, we can select S_1 and S_2 . If S_1 decreases and S_2 increases, fault f_1 is detected; If S_1 decreases and S_2 keeps constant, fault f_2 is detected. If $t_2 < t_1$, we have no problem to identify each fault.

However, if $t_1 < t_2$ and fault f_1 occurs, between time t_1 and time t_2 , S_1 decreases and S_2 is constant, fault f_2 will be claimed. This is of course a false positive alarm. Therefore, the following claim can be easily proved.

Claim 4.2: Assume sensors in the set $A_k = \{S_{i1}, S_{i2}, \dots, S_{im}\}$ are selected to detect fault k without considering the sensor directions, and sensors in the set $A_i = A_k \cup \{S_{k1}, S_{k2}, \dots, S_{kq}\}$ are selected fault i without considering the sensor directions. Also, if fault i occurs, the sensors in the set A_k are affected earlier than any other sensors. In this case, if fault k occurs, a false positive alarm of fault k will be fired.

Proof of claim 4.2: if fault i occurs, all the sensors in A_k are affected first, therefore, fault k will be claimed, and a false positive alarm will be fired.

To reduce this type of false alarm, a penalty in the objective function can be applied if this situation occurs.

4.2.4 Fault Detection Reliability (R)

When a fault occurs, it may not be detected because of the failure of sensors. Fault detection reliability is defined as the probability that a fault is detected when it occurs. If a fault is detected by sensors that are prone to failure, then the fault detection reliability is low. In a system, each sensor has a Probability of Failure (PF). If sensors S_1, \dots, S_q are selected to detect fault f_i , and number of each sensor is nn_1, \dots, nn_q ; then the probabilities of failure are FR_1, \dots, FR_q , respectively. Let FR_i be the probability that a fault f_i is detected when it occurs. This is detection reliability of fault f_i , which is given by:

$$R_i = \frac{\sum_{j=1}^q (1 - FR_j^{m_j})}{q} \quad (4-12)$$

where

FR_j : probability of failure of sensor j.

m_j : the number of sensor j.

4.2.5 Limited Resources

In some situations, sensor localization cost needs be taken into account because of limited resources. Sensor localization cost includes sensor price, cost induced by installation, data processing, data fusion, and others.

4.3 FMECA Study

A FMECA study identifies failure modes and specifies the severity of failures and frequency of their occurrence. When the consequence of a fault is more severe, or the occurrence is more frequent, higher detectability and fault detection reliability requirements must be imposed. In other words, detectability, identifiability, and fault detection reliability requirements are determined by the criticality and frequency of an occurrence of a fault based on the FMECA study.

On the basis of the FMECA study, failure modes are classified according to their severity and frequency of occurrence.

S: severity index; an assessment of how serious the effect of the potential failure mode is.

O: frequency of occurrence; an assessment of the likelihood that a particular fault will occur and result in a failure mode during the intended life and use of the product.

The severity index categorizes a failure mode according to its ultimate consequence.

The severity index is defined on a scale from 1 to 4, with 4 being the most severe:

1 — Minor: a failure that does not cause injury or equipment damage, but may result in equipment failure if left unattended, down time, or unscheduled maintenance/repair.

2 — Marginal: a failure that may cause minor injury, equipment damage, or degradation of system performance.

3 —Critical: a failure that may cause severe injury, equipment damage, and termination

4 — Catastrophic: a failure that results in death, significant injury, or total loss of equipment.

The frequency of occurrence can be defined on a scale from 1 to 4 (or possibly more divisions) with 1 being the lowest probability to occur. The classification number is derived based on failure occurrence for the particular event standardized to a specific time period and broken down into unlikely, occasional, probable, and likely. For example, with a Mean Time Between Failure (MTBF) range of 1000 hours, the classification number is

1 — Unlikely: cumulative failures less than 1.0

2 — Occasional: cumulative failures from 1.0 to 10

3 — Probable: cumulative failures from 10 to 100

4 — Likely: cumulative failures greater than 100

4.4 The Modeling Approach

Fault propagation information is necessary and important for sensor localization/selection at the system level. Different approaches, such as Petri nets, fault

trees and digraph-based methods are available to model the fault propagation. Among them, the Directed Graph (DG) and Signed Directed Graph (SDG) methods, with simple graphical representations of a system that represent the cause-effect analysis of fault propagation, are widely used [9][23]. A DG model of a process consists of a set of nodes and directed branches. The nodes represent process variables and the branches represent the causal influences between the nodes. There is a branch connecting one node to another if the first node affects the latter. In a SDG model, each branch has a sign (+ or -) associated with it, which indicates whether the cause and effect variables tend to change in the same (+) or opposite (-) direction [53]. The SDG of the process can be built from the process equations that are used to model the process [54].

Neither a DG model nor a SDG model includes any quantitative information related to fault diagnosis, i.e., how large the Signal-to-Noise Ratio (SNR) at a sensor node is and how sensitive and how fast a sensor node can measure a fault. The quantitative information can be acquired by testing the system thoroughly, which is expensive in most cases. A Quantified Directed Graph (QDG) model is proposed to simplify such a test.

In a QDG model, each node stands for a possible sensor location and is assigned a SNR in dB with it. Each arc has a “+” or “-” sign associated with it, as well as the fault propagation time and the fault propagation gain between the two nodes connected by the arc. The fault propagation time and the propagation gain are defined based on the step response that is defined as the response of a system to a step input. The fault propagation time (PT) is defined as the rise time when the response reaches 10% of the steady gain, while the fault propagation gain (PG) is defined as the steady-state gain.

Type and Location of Possible Sensors

Based on the QDG model, possible type and possible location of sensors can be easily determined by scanning all the sensors in the fault propagation path.

Sensor Fault Detectability

If a propagation path exist from fault i to sensor j , and the propagation gains along the path are $\{PG_{j1}, PG_{j2}, \dots, PG_{jn}\}$, then the propagation gain of this path can be calculated by multiplying all the propagation gains along the path, which is $\prod_{k=1}^n PG_{jk}$. If the

propagation time along the path are $\{PT_{j1}, PT_{j2}, \dots, PT_{jn}\}$, then the time-to-detection of sensor j to fault i is estimated as the summation of the propagation time along this path,

which is $\sum_{k=1}^n PT_{jk}$

If multiple propagation paths exist from fault i to sensor j , then the overall propagation gain from the fault i to sensor j is defined as the summation of all the propagation gains of each path, and the time-to-detection of sensor j to fault i is defined as the minimum propagation time of all the paths. The symptom duration at sensor j is calculated as the maximum symptom duration at sensor j for each path.

Therefore, sensor fault detectability can be calculated using the QDG model. The QDG model is validated using an example.

Example 4.2. Let there be three sensor nodes A, B, and C, and two faults f_1 and f_2 . Assume the SNR at sensor nodes A, B, and C are -2dB, 15dB, 10dB, respectively and all the sensor resolutions are 1.

Fault f_1 : assume it takes 2 minutes to be measured at sensor node A after fault f_1 is initiated; 25 minutes to propagate from A to B, 2 minutes to be measured at sensor node C; the fault detection sensitivity at sensor nodes A and C is 0.9; the propagation gain

from A to B is 1.0; the symptom duration measured at sensor nodes A, B, and C is equivalent to the time-to-failure, which is 100 minutes.

Fault f_2 : assume it takes 2 minutes and 100 minutes to be measured at sensor nodes B and C after f_2 is initiated, respectively; the fault detection sensitivity at sensor nodes B and C is 0.9; the symptom duration measured at sensor nodes A, B, and C is equivalent to the time-to-failure, which is 100 minutes.

The resultant QDG model is shown in Figure 15. The fault detection sensitivity for fault f_1 at nodes A, B, and C can be calculated from the QDG model: $V_{1A} = 0.9$; $V_{1B} = 0.9$ *1=0.9; $V_{1C} = 0.9$.

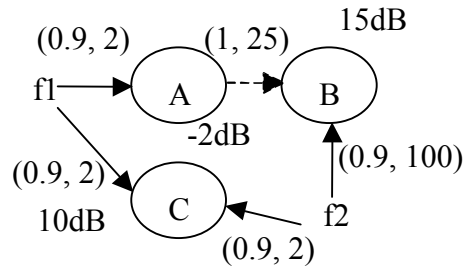


Figure 15: A QDG example.

The ratio of the time-to-detection to the time-to-failure at each node A, B, and C, is $(TTD/TTF)_{1A} = 2 / 100 = 0.02$, $(TTD/TTF)_{1A} = (2+25) / 100 = 0.27$, and $(TTD/TTF)_{1A} = 2 / 100 = 0.02$, respectively.

Meanwhile, $V_{2B} = 0.9$; $V_{2C} = 0.9$; $(TTD/TTF)_{2B} = 100/100 = 1$; $(TTD/TTF)_{2C} = 2/100 = 0.02$.

The sensor detectability of faults f_1 and f_2 for each sensor can be calculated as: $SD_{1A} = 0.07$, $SD_{1B} = 0.83$, $SD_{1C} = 0.96$, $SD_{2A} = 0$, $SD_{2B} = 0$, and $SD_{2C} = 0.96$.

4.5 FOM

The proposed FOM maximizes the fault detectability and minimizes the required number of sensors while achieving optimum sensor localization/selection. The FOM is in the form of the weighted sum of the fault detectability and the number of sensors. The weights are adjustable and are mainly determined by the severity of failure effects and the probability of failure occurrence.

The following constraints apply:

- All faults can be detected, i.e. the detectability of each fault is greater than 0;

A binary matrix B is generated to indicate the sensor-fault mapping. The rows of the matrix B correspond to the faults, while the columns correspond to the sensors. Entry b_{ij} is 1 if sensor j is selected to detect fault i ; otherwise, b_{ij} is 0. Thus, the total fault detectability of fault i is calculated by

$$D_i = \frac{\sum_{j=1}^M b_{ij} * SD_{ij}}{\sum_{j=1}^M b_{ij}} \quad (4-13)$$

- The fault detectability for selected faults is required to be greater than predefined low bounds that are determined by the fault severity and the probability of occurrence;
- The fault detection reliability for selected faults is required to be greater than predefined low bounds that are determined by the fault severity and the probability of occurrence;

- The resource may be limited explicitly ($\leq C^*$). Let the cost of a sensor j be C_j , including its price and other cost induced by installation, data processing, data fusion, and maintenance. To simplify the case, it is assumed that the cost to install any additional sensor j is the same as to install the first sensor j . If the number of sensor j is X_j , then

$$\sum_{j=1}^M C_j * X_j \leq C^* \quad (4-14)$$

Also, let the maximum possible number of sensors be M_{\max} , then the constraint of number of sensors can be written as:

$$\begin{aligned} \sum_{j=1}^M X_j + C_s &= M_{\max} \\ C_s &\geq 0 \end{aligned} \quad (4-15)$$

In the above formulation, C_s is a slack variable, which takes nonnegative integer values. Thus, to minimize the number of sensors ($\sum_{j=1}^M X_j$) is equivalent to maximizing the slack variable C_s .

- All faults can be identified, i.e., the detectability of each virtual fault is greater than 0.
- The number of sensors at selected locations may be limited explicitly.
- A penalty is applied if the symptom sequence requirement is not satisfied.

In summary, the FOM can be written in the following formula:

$$\left\{ \begin{array}{l}
Max \left\{ \sum_{i=1}^{N'} K_i * D_i + \alpha * C_s - \beta * \sum_{i=1, j=1}^{i=N, j=N} PFP_{ij} \right\} \\
s.t. \\
D_i = \frac{\sum_{j=1}^M b_{ij} * SD_{ij}}{\sum_{j=1}^M b_{ij}} > 0 \quad i = 1, 2, \dots, N' \\
D_i \geq D_i^* \quad \forall i \in M_D \\
R_i = \prod_{j=1}^M (1 - (b_{ij} * FR_j)^{X_j}) \quad i = 1, 2, \dots, N' \\
R_i \geq R_i^* \quad \forall i \in M_R \\
\sum_{j=1}^M C_j * X_j \leq C^* \\
\sum_{j=1}^M X_j + C_s = M_{\max} \\
X_j \leq N_j \quad j = 1, \dots, M \\
X_j \geq 1 \Leftrightarrow \sum_{i=1}^N b_{ij} \geq 1 \quad j = 1, \dots, M \\
C_s \geq 0
\end{array} \right\} \quad (4-16)$$

Where,

K_i : weight of detectability decided by fault severity & occurrence;

SD_{ij} : the sensor detectability of fault f_i for sensor s_j ;

D_i : detectability of fault f_i ;

α : weight of the cost of sensor j ;

β : penalty weight;

PFP_{ij} : penalty of false positive alarm;

α : weight of the cost of sensor j ;

C_s : slack variable of number of sensors;

C_j : cost of sensor j .

C^* : resource limit;

b_{ij} : binary variable indicating the sensor-fault relationship: 1 if sensor j is selected to detect fault i , otherwise, 0;

X_j : number of sensor at sensor location j .

D_i^* : low bound of detectability of fault f_i decided by fault severity & occurrence;

N : number of original faults;

M : number of original sensors;

N' : number of faults, including “virtual faults”;

R_i : fault detection reliability of fault f_i ;

R_i^* : low bound of fault detection reliability of fault f_i decided by fault severity & occurrence;

M_R : the fault set that has fault detection reliability requirement;

M_D : the fault set that has fault detectability requirement;

FR_j : the probability of failure of sensor j ;

N_j : the maximum sensors that can be put at location j .

Any other requirement can be added easily as a constraint. Therefore, this FOM is flexible expandable.

4.6 Optimization

The optimization step includes two main tasks: optimize sensor locations and optimize selected sensors. The selected FOM is an Integer Nonlinear Programming (INLP) model.

Current commercial INLP solvers, such as SBB and DICOPT, can be used to find the optimal number of sensors. In the optimization module, not long the number of sensors at each location needs to be optimized, but also the sensors selected for each fault needs to be determined. To detect and identify N fault modes by placing sensors at M possible sensor locations, in the worst case, the maximum optimization variables will be $M*N(1+(N-1)/2)$. For example, $N = 50$ and $M = 100$, the maximum number of variables to be optimized becomes 127500!

To reduce the optimization scale, an optimization algorithm is proposed in this research to optimize the formularized Figure-of-Merit. This algorithm combines particle swarm optimization method with a heuristic search algorithm. The possible number of sensor locations is optimized based on particle swarm optimization method and the sensors selected for each fault is selected based on a heuristic search algorithm.

Sensor Localization

PSO method is originally designed for continuous variables optimization with the following format:

$$\begin{aligned}
 &\min\{f(X)\} \\
 &s.t. \\
 &X \in R^n \\
 &X^L < X < X^H
 \end{aligned}
 \tag{4-17}$$

The formulized FOM has discrete variables and both equality and inequality constraints. We need to convert the INLP problem to a standard PSO problem.

Penalty functions are usually used to convert the constraint optimization problem to an unconstrained problem. Since the detectability requirements are more important to the

reliability requirements in most cases, the penalty for breaking the detectability requirements are set to be larger than breaking the detection reliability requirements.

The number of sensors at each location is a discrete variable; therefore, the real number variables need to be rounded to integer numbers.

Therefore, the original INLP problem can be rewritten as the following equation.

$$\left\{ \begin{array}{l} \text{Max} \left\{ \sum_{i=1}^N K_i * D_i + \alpha * C_{s_s} - \beta * \sum_{i=1, j=1}^{i=N, j=N} PFP_{ij} + \text{Penalty}D * (D_i - D_i^*) + \text{Penalty}R * (R^i - R_i^*) \right\} \\ \text{s.t.} \\ \sum_{j=1}^M X_j + C_s = M_{\max} \\ X_j \leq N_j \quad j = 1, \dots, M \end{array} \right\} \quad (4-18)$$

Each particle is an integer array indicating the number of sensors selected at each location: $\{S_1, S_2, \dots, S_n\}$. PSO is initialized with a group of random particles and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution it has achieved so far. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest [64].

After finding the two best values, the particle updates its velocity and positions with following equations.

$$v[] = w * v[] + c1 * \text{rand}() * (\text{pbest}[] - \text{present}[]) + c2 * \text{rand}() * (\text{gbest}[] - \text{present}[]) \quad (4-19)$$

$$\text{NewNumber}[] = \text{present}[] + v[] \quad (4-20)$$

$v[]$ is the particle velocity, $persent[]$ is the current particle (solution). $pbest[]$ and $gbest[]$ are defined as stated before. $rand()$ is a random number between (0,1). $c1$, $c2$ are learning factors. Usually $c1 = c2 = 2$.

The pseudo code of the procedure is as follows.

Initialize matrix $B0 = \{b_{ij}\}$.

$b_{ij} = 1$ if the sensor fault detectability of sensor j to detect fault i is greater than 0;

$b_{ij} = 0$; otherwise.

For each particle

Initialize particle based on the initial matrix $B0$

END

Do

For each particle

Calculate fitness value

*If the fitness value is better than the best fitness value ($pBest$) in history
set current value as the new $pBest$*

End

Choose the particle with the best fitness value of all the particles nearby as the $gBest$

For each particle

Calculate particle velocity according equation

Update particle position according equation

End

While maximum iterations or minimum error criteria is not attained

Particles' velocities on each dimension are clamped to a maximum velocity V_{\max} . If the sum of accelerations would cause the velocity on that dimension to exceed V_{\max} , which is a parameter specified by the user. Then the velocity on that dimension is limited to V_{\max} .

Sensor Selection

At each iteration for each particle, an important issue is how to select sensors for each fault to achieve maximum FOM based on the sensors selected, i.e., determine matrix $B=\{b_{ij}\}$ based on selected sensor set $A=\{S_1, S_2, \dots, S_n\}$, with the minimum detectability $D_{\min}(i)$, minimum reliability $R_{\min}(i)$, and other requirements. A heuristic search algorithm is proposed. For each fault:

- 1) All the sensors in the sensor set A are sorted by the sensor fault detectability in a descendent order. Assume the sorted sensor set is $\{S_1, S_2, \dots, S_n\}$, S_1 with the highest sensor fault detectability. Check each possible sensor for detectability and reliability requirements, and find the sensor with maximum detectability that meets all the constraints. If several sensors have the same detectability, select the one with maximum reliability; if no such sensor exists, the constraints can not be met and penalty will apply, go to step 3. Let the selected sensor be S_k . Selected number of sensor location is set to 1: $ns=1$. Let the index $I = k$. Calculate the current fault detectability $D(i)$ and detection reliability $R(i)$. Construct a new sensor set $SS: \{S_j | j < I\}$.
- 2) Try to find another sensor in the set with detectability greater than the current fault detectability $D(i)$ and the probability that it will not fail is no less than $(R_{\min}(i) - ns * (R(i) - R_{\min}(i)))$. If such a sensor is found, calculate the new

detection reliability as the mean value of the probability that the selected sensors

will not fail: $R_i = \frac{\sum_{j=1}^{ns+1} 1 - FR_j^{m_j}}{ns + 1}$ and new detectability as the mean value of the

detectability of selected sensors: $D_i = \frac{\sum_{j=1}^{ns+1} D_j}{ns + 1}$. If no such a sensor exists, go to the

next step; otherwise, $ns := ns + 1$, remove the selected sensor from the sensor set SS, and repeat this step.

- 3) Calculate current Figure-Of-Merit based on the selected sensors.

Consider the example 4.2 again, it is assumed that all the detectability weights in the FOM are equal to 1 and the cost weight is -1. Additionally no resource limits or low bounds of detectability and reliability exist. A virtual fault f_3 is created to distinguish between faults f_1 and f_2 . Sensors B and C are selected after solving the INLP model. The detectability of faults f_1 , f_2 , and the virtual fault f_3 is 0.9, 0.96, and 0.83, respectively.

If the DG model approach is used based on the algorithm proposed by [9] whose purpose is to minimize the number of sensors, sensors A and B (or A and C) are selected to detect and identify faults f_1 and f_2 . If sensors A and B are selected, the fault detectability of fault f_2 is 0 since $SD_{2A} = 0$ and $SD_{2B} = 0$. If sensors A and C are selected, sensor A is expected to distinguish fault f_1 from fault f_2 . However, it is difficult to distinguish between them since SD_{1A} and SD_{2A} are both very small.

If the SDG model approach is used [23], sensor B is selected to detect and identify faults f_1 and f_2 . However, fault f_2 is not able to be detected by sensor B since $SD_{2B} = 0$.

Therefore, the proposed sensor localization/selection approach based on the QDG model is able to detect faults with high detectability and a small number of sensors.

4.7 Performance Assessment

The performance of the sensor localization/selection strategy is validated and verified in the integrated diagnostic/prognostic architecture.

The first goal of the performance module is to estimate the fault detection error rate with selected sensor suite. For a fault that is concerned, it may not be detected by selected sensors, or the selected sensors may claim other faults happening by mistake. Therefore, for each fault, Single False Positive Rate (SFPR) and Single False Negative Rate (SFNR) are defined as the performance metrics.

- Single False Positive Rate (SFPR)

$$SFPR = \frac{\text{Number of misfired faults}}{\text{Number of faults interested}} \quad (4-21)$$

- Single False Negative Rate (SFNR)

$$SFNR = \begin{cases} 1 & \text{if the fault is detected} \\ 0 & \text{if the fault is not detected} \end{cases} \quad (4-22)$$

Based on the definitions of SFPR and SFNR of each fault, the Average False Positive Rate (AFPR) and the Average False Negative Rate (AFNR) are defined in the flowing equations.

- AFPR

$$AFPR = \frac{\sum SFPR}{\text{Number of faults interested}} \quad (4-23)$$

- AFNR

$$AFNR = \frac{\sum SFNR}{\text{Number of faults interested}} \quad (4-24)$$

To estimate how fast a fault is detected after it is initiated, time delay is defined.

- Time Delay (TD) – time span between the initiation and the detection/isolation/identification of a fault (failure) event.

$$TD = T_d - T_i \quad (4-25)$$

Where,

T_d : fault (failure) detected/identified/isolated time;

T_i : detection initiation time.

If the performance is not satisfactory, i.e., the SFPR, SFNR, AFPR, AFNR, or time delay does not meet pre-defined specifications, the sensor suite may not be well selected for fault diagnosis. Sensors needs to be re-selected, and the following steps can be taken, which is also shown in Figure 16:

- The detectability weights can be adjusted in the FOM accordingly.
- The Detectability and the reliability definitions can be adjusted according to the diagnostic performance information. The original definitions of detectability and reliability can be revised as follows:

$$D_i = K_D * \frac{\sum_{j=1}^m SD_{ij}}{m}; \quad R_i = K_R * \prod_{j=1}^q (1 - FR_j^{nn_j})$$

where

FR_j : probability of failure of sensor j.

nn_j : the number of sensor j.

K_D and K_R vary from 0 to 1 based on the feedback performance information.

- By going back to the FMECA study level, the fault propagation and fault cause-effect relationships can be re-examined.

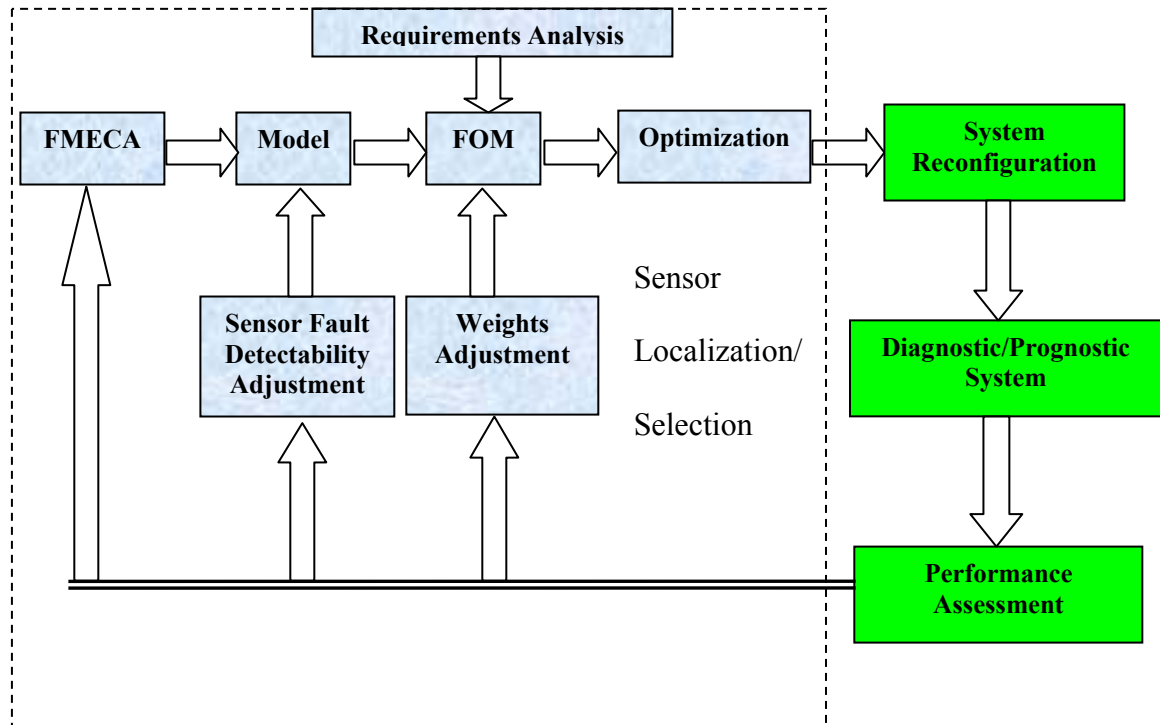


Figure 16: Performance Assessment Feedback

CHAPTER 5

APPLICATION EXAMPLES

5.1 Process Demonstrator

The proposed integrated diagnostic/prognostic architecture is applied to a three-tank process demonstrator where faults are seeded and detected, and the remaining useful lifetime of the failing component is calculated.

The process demonstrator is a model of a continuous fluid process typical of those found in industry, as shown in Figure 17. The demonstrator consists of tanks, pipes, valves, pumps, mixers, and electric heaters. By activating these devices, fluid can be stirred, heated, and circulated among the tanks. Honeywell Smart Transmitters installed in the system monitor fluid level, fluid flow and fluid temperature.

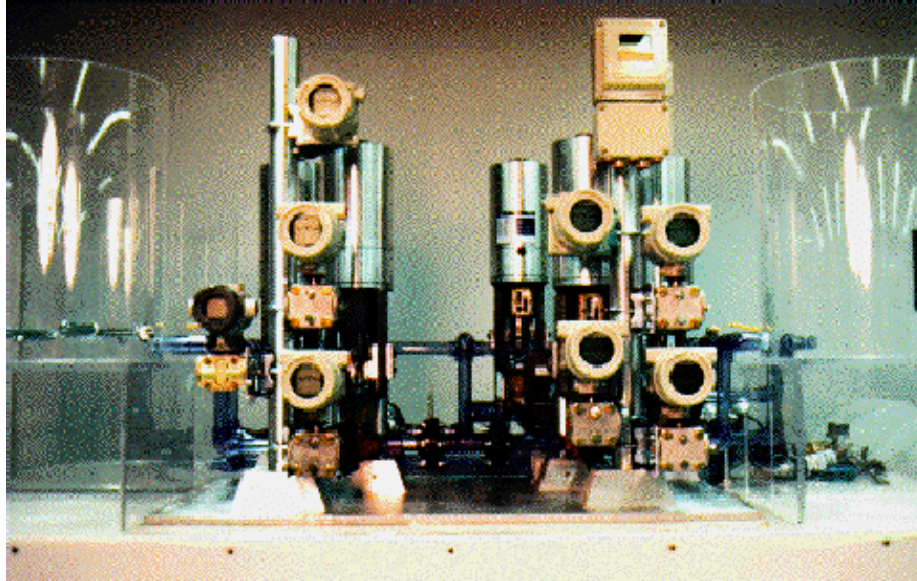


Figure 17: A process demonstrator.

There are three tanks in the system: two 25-gallon tanks and one 50-gallon tank. Each tank has a pump on the output line. There are flow valves and flow meters on the lines into and out of tanks 1 and 2. In addition, there is a flow valve and flow meter on the line from tank 1 to tank 2. There are hand valves into tanks 1 and 2 for introducing disturbances into the system. Tank 1 and Tank 2 each have a mixer to stir the fluid in the tank.

A Labview interface of the process demonstrator for data collection and control purposes is shown in Figure 18.

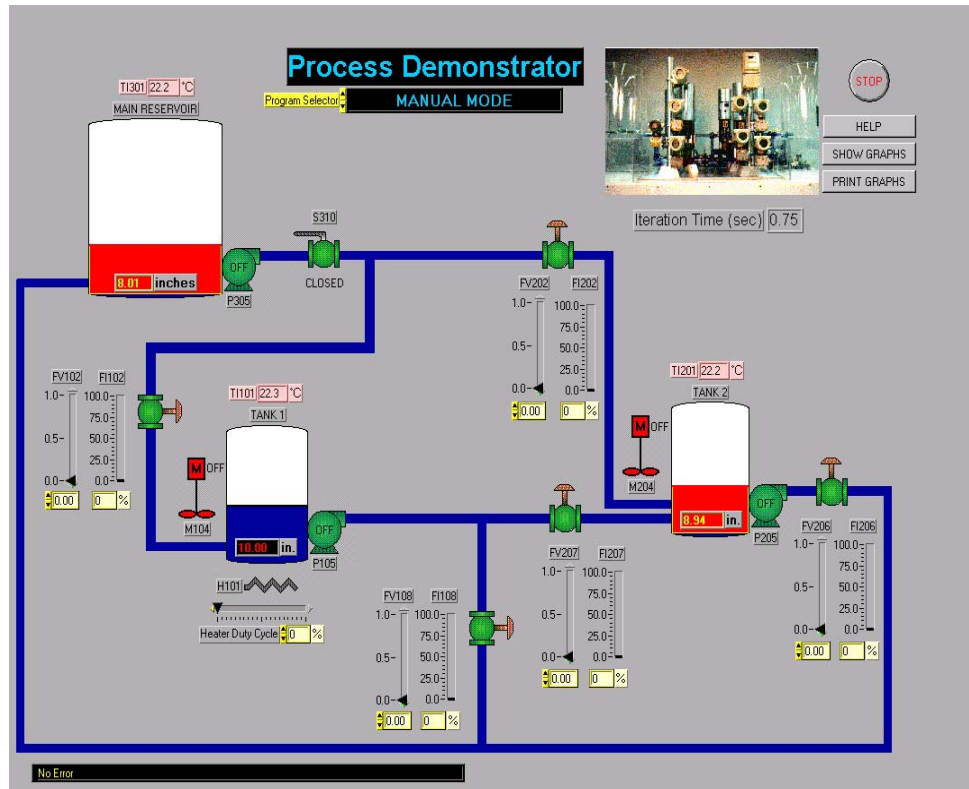


Figure 18: Process demonstrator diagram in Labview.

5.1.1 Problem Statement

To simplify the case, two operating modes are detected, “idle” mode and “pump-on” mode.

In the “idle” operating mode, the pumps are off and the valves are in their closed positions. Only a fault, “Tank 1 leakage”, is considered under this operating mode.

In the “pump-on” mode, water is pumped from Tank 1 to main reservoir continuously. Under this mode, a “Tank 1 level low” fault is claimed when the Tank 1 level is lower than 8 inches.

The fault modes, operating modes, and associated features are summarized in Table 2.

Table 2: Failure modes, operating modes, and features.

Failure Mode	Feature 1	Feature 2	Feature 3	Operating Mode
Tank1 leakage	Tank 1 level slope	Tank 2 level slope	Tank 1 level	Transfer water from Tank 1 to Tank 2
Tank 1 leakage	Tank 1 level slope	Tank 1 level		System idle
Tank 1 level low	Tank 1 level			Transfer water from Tank 1 to Tank 2
Tank 1 level low	Tank 1 level			System idle

5.1.2 Diagnostic/Prognostic Results

Following these two operating modes, “idle” and “pump on” modes, two sets of classifiers and prognosticators were employed to reduce the system resource consumption and speed up the neural network training and prediction process.

The user interface is shown in the Graphical User Interface (GUI, Figure 19) and the diagnostic result is shown in Figure 20.

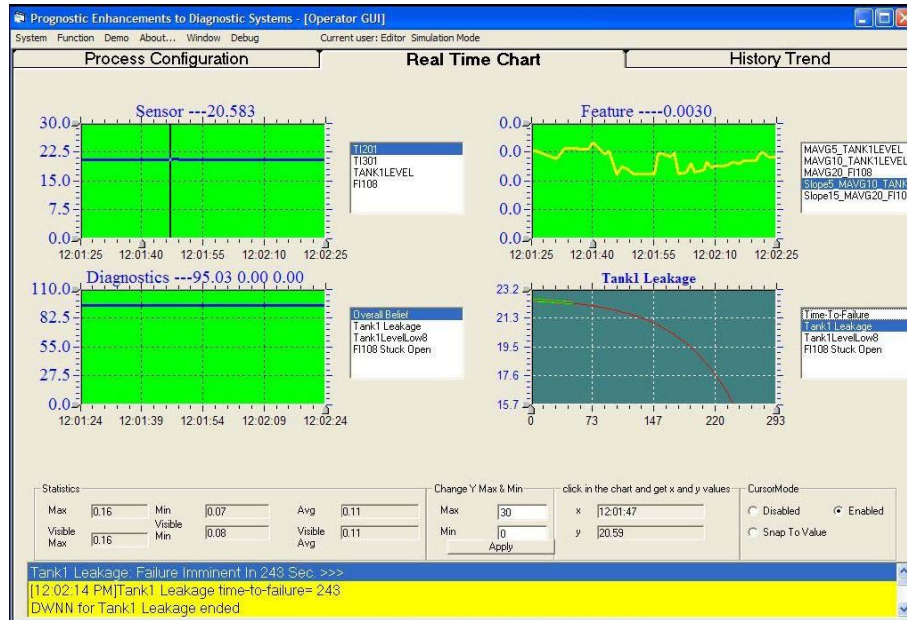


Figure 19: Graphical user interface.

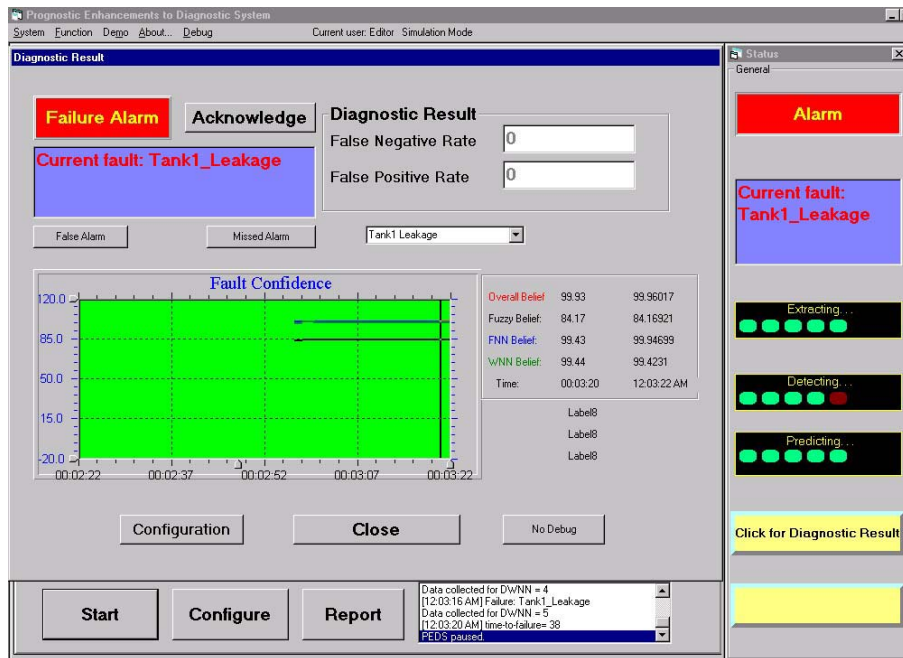


Figure 20: A Diagnostic Result example.

Table 3 is an example of the diagnostic results from individual diagnostic module and an overall belief calculated by the Dempster-Shafer theory.

Table 3: Fault belief of Tank1 leakage.

Fuzzy	WNN	AFNN	Overall
84.17	99.44	99.43	99.93

After a fault is detected, the DWNN module is called to read the fault information from the database and predict the remaining useful lifetime of the failing component, as shown in Figure 21. In the figure, the green line is the sampled Tank 1 level data from the demonstrator, and the yellow line gives the prediction at that time. An example of the prognostic result analysis for the Tank 1 leakage fault is given in Table 4.

Table 4: Prognostic results analysis of Tank 1 leakage.

Prediction Given Time Step	32	51	70
Predicted Failure Time Step	77	66	63

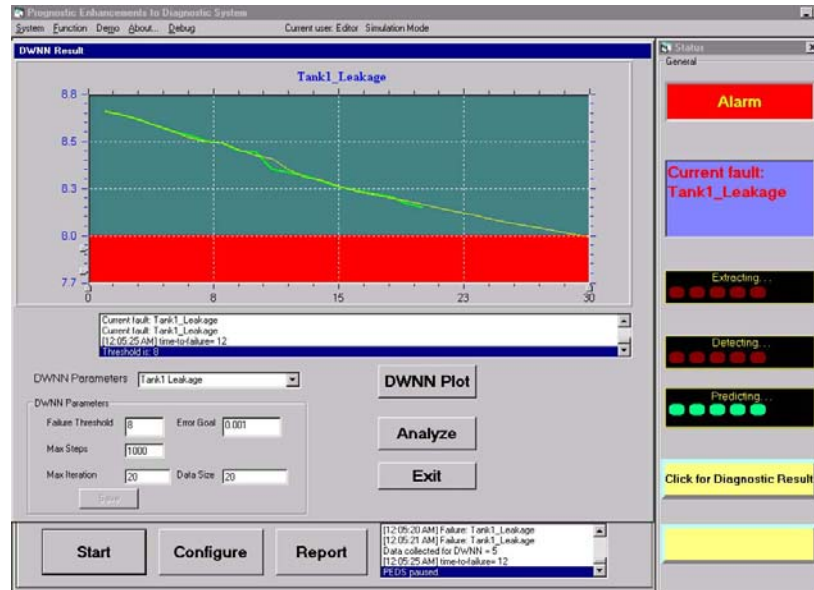


Figure 21: A prognostic result example.

The failure occurs at time step 63. It can be seen that the prediction is updated with more incoming data.

5.1.3 Diagnostic Performance Assessment

Ten sets of data were collected from the Process Demonstrator for performance assessment. These 10 data sets were split into two groups; the first group includes 3 data sets used for training purpose, while the second group including 7 data sets was used to validate the trained parameters. The following tables list the diagnostic performance analysis results for different diagnostic modules: Fuzzy, WNN, AFNN, and Dempster-Shafer.

Table 5: Fuzzy results.

Fuzzy					
Data Source	Threshold	FPR (%)	FNR (%)	FAR (%)	TTD (s)
D0923151	0.8	9.62	2.89	87.5	3
D0923152	0.8	0	3.96	96.04	16
D0923153	0.8	6.67	2.86	90.48	12
D0923161	0.8	4.90	3.92	91.18	15
D0923162	0.8	0	9.90	90.10	19
D0523	0.8	0	9.26	90.74	19
D061203151	0.8	0	1.01	98.99	7
D061203152	0.8	0	10.58	89.42	17
D061203154	0.8	0	20.41	79.59	21

Table 6: WNN results.

WNN					
Data Source	Threshold	FPR (%)	FNR (%)	FAR (%)	TTD (s)
D0923151	0.6	2.89	2.89	94.23	3
D0923152	0.6	0	3.96	96.04	16
D0923153	0.6	3.81	4.76	91.43	15
D0923161	0.6	0	3.92	96.08	15
D0923162	0.6	0	10.89	89.11	19
D0523	0.6	0	12.04	87.96	23
D061203151	0.6	0	1.01	98.99	7
D061203152	0.6	0	11.54	88.46	17
D061203154	0.6	0	21.43	78.57	24

Table 7: AFNN results.

AFNN					
Data Source	Threshold	FPR (%)	FNR (%)	FAR (%)	TTD (s)
D0923151	0.9999	8.65	2.89	88.46	3
D0923152	0.9999	0	3.96	96.04	16
D0923153	0.9999	6.67	3.81	89.52	12
D0923161	0.9999	4.90	3.92	91.18	15
D0923162	0.9999	0	10.89	89.11	19
D0523	0.9999	0	10.19	89.81	19
D061203151	0.9999	0	1.01	98.99	7
D061203152	0.9999	0	10.58	89.42	17
D061203154	0.9999	0	20.41	79.59	21

Table 8: Dempster-Shafer results.

Overall					
Data Source	Threshold	FPR (%)	FNR (%)	FAR (%)	TTD (s)
D0923151	0.95	5.77	2.89	91.35	3
D0923152	0.95	0	3.96	96.04	16
D0923153	0.95	4.76	4.76	90.48	15
D0923161	0.95	1.96	3.92	94.12	15
D0923162	0.95	0	10.89	89.11	19
D0523	0.95	0	11.11	88.89	19
D061203151	0.95	0	1.01	98.99	7
D061203152	0.95	0	10.58	89.42	17
D061203154	0.95	0	20.41	79.59	21

The minimum time-to-detection is 3 seconds, which is the time to read the sensor data from the database, extract the features, and classify the results. And the time-to-failure for the leakage fault is about 3600 seconds. Therefore, the time-to-detection to time-to-failure is small for all the results. Also, the FPRs and FNRs are small except the data set of “D061203154.” The high FNR for this data set (around 20%) is due to the leakage is much smaller than others.

5.1.4 Prognostic Performance Assessment

To validate and verify the performance of prognostic modules, six sets of faulty data were used from the Process Demonstrator. For each data set, a leakage fault is initiated and evolved to a failure. For each data set, the prediction results converge to the real failure time, which is the time span between a fault is initiated and a failure occurs, as shown in Figure 22.

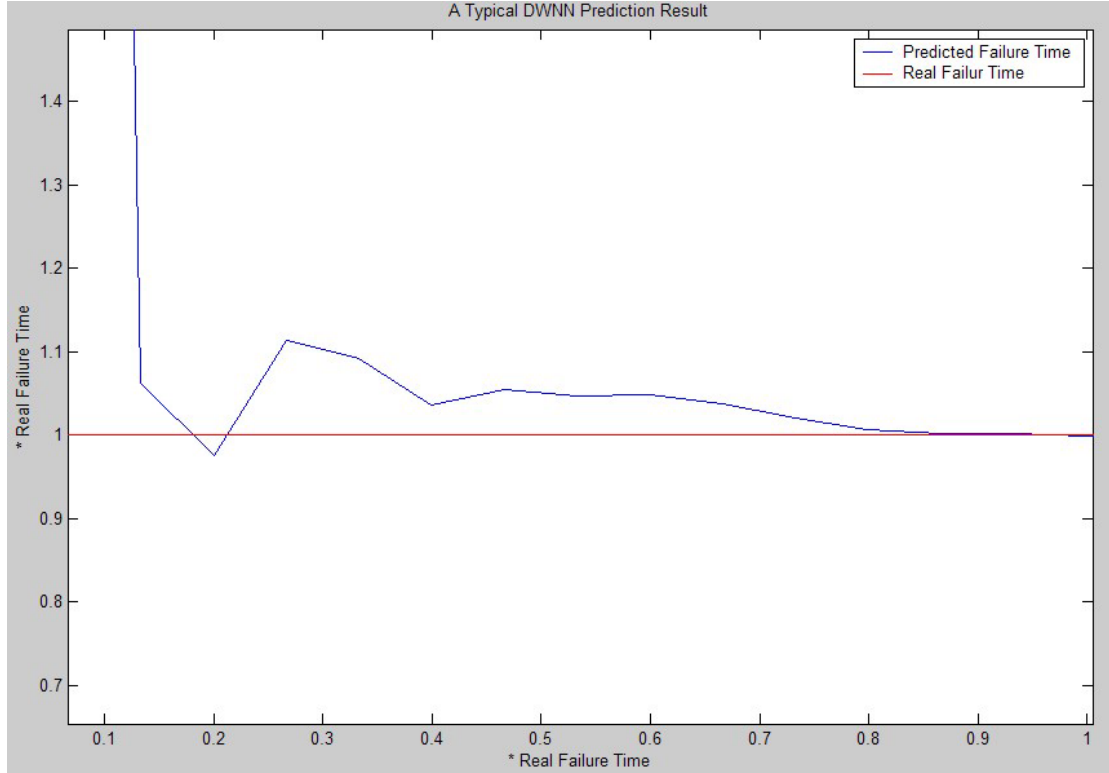


Figure 22: DWNN prediction results.

We evaluate the prognostic performance from two aspects: horizontal and vertical. Vertical Metrics are defined to assess the prognostic performance for a single fault propagation data:

- Rising Time: the time when the predicted time-to-failure reaches 80 percent of the real time-to-failure.
- Settling Time: the time when the error between predicted time-to-failure and the real time-to-failure is less than 20% of the real time-to-failure.

These metrics are evaluated in the following table. In this table, the rise time and the settling time are calculated and are normalized using the real failure time.

Table 9: Rise time and settling time analysis.

Data Set	Rise Time (Seconds)	Rise Time/ Failure Time	Settling Time (Seconds)	Settling Time/ Failure Time	Failure Time (Seconds)
D05060416	100	6.67%	200	13.32%	1502
D05060417	500	34.82%	700	34.82%	1436
D05060418	100	7.33%	300	21.99%	1364
D05060419	100	6.61%	200	13.23%	1512
D05070410	100	4.81%	600	28.86%	2079
D05070412	100	2.53%	2000	50.67%	3947
Mean		10.46%		27.15%	

From the results given in the table, the mean ratio of rise time to real failure time is 10.46%, and the mean ratio of settling time to Failure Time is 27.15%.

Horizontal Metrics are based on repeated fault propagation data and statistical prediction results. Accuracy is commonly used as a horizontal metrics [66].

In Table 10, the accuracy is calculated at the following time intervals: 1/10 Failure Time, 2/10 Failure Time... 9/10 Failure Time.

Table 10: Accuracy results.

Data	Exp(-D _i (t)/D ₀): t = Failure Time *								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
D0506-0416	0.939	0.976	0.893	0.911	0.954	0.952	0.979	0.993	0.998
D0506-0417	0.526	0.754	0.817	0.885	0.911	0.919	0.928	0.945	0.975
D0506-0418	0.367	0.867	0.919	0.947	0.959	0.964	0.992	0.998	0.997
D0506-0419	0.936	0.981	0.947	0.970	0.986	0.998	0.988	0.988	0.994
D0507-0410	0.000	0.695	0.819	0.906	0.939	0.943	0.976	0.975	0.984
D0507-0412	0.000	0.000	0.000	0.641	0.838	0.927	0.978	0.998	0.998
Mean/ Accuracy	0.461	0.712	0.732	0.877	0.931	0.950	0.973	0.983	0.991

The accuracy results are plotted in the following figure.

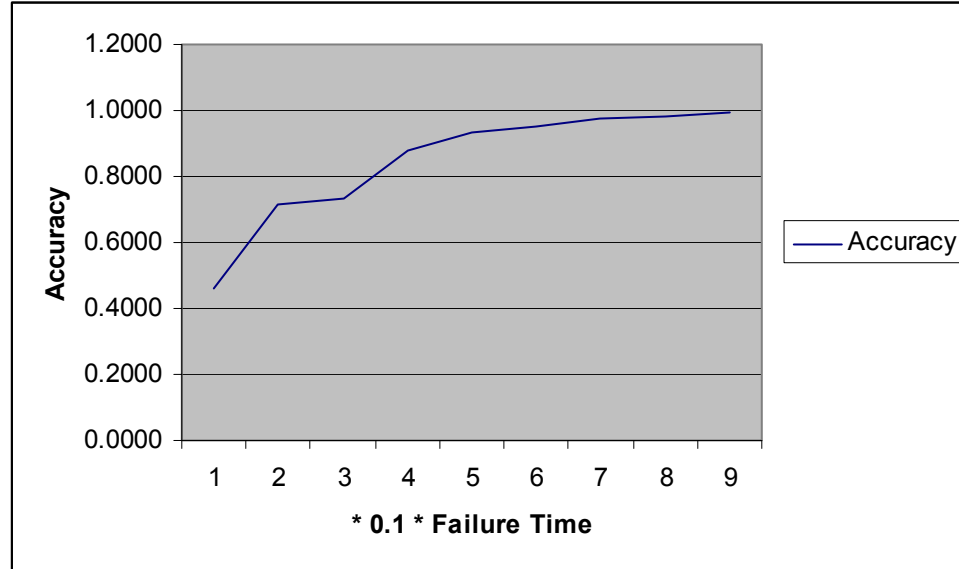


Figure 23: Accuracy analysis.

As we can see from the above plot, the accuracy is approaching to 1 with continuous prediction.

5.2 Optimum Sensor Placement

5.2.1 Problem Statement

A more complex five-tank system (originally from Chang *et al.* [56]) is employed to demonstrate the proposed sensor localization/selection approach, which is shown in Figure 24.

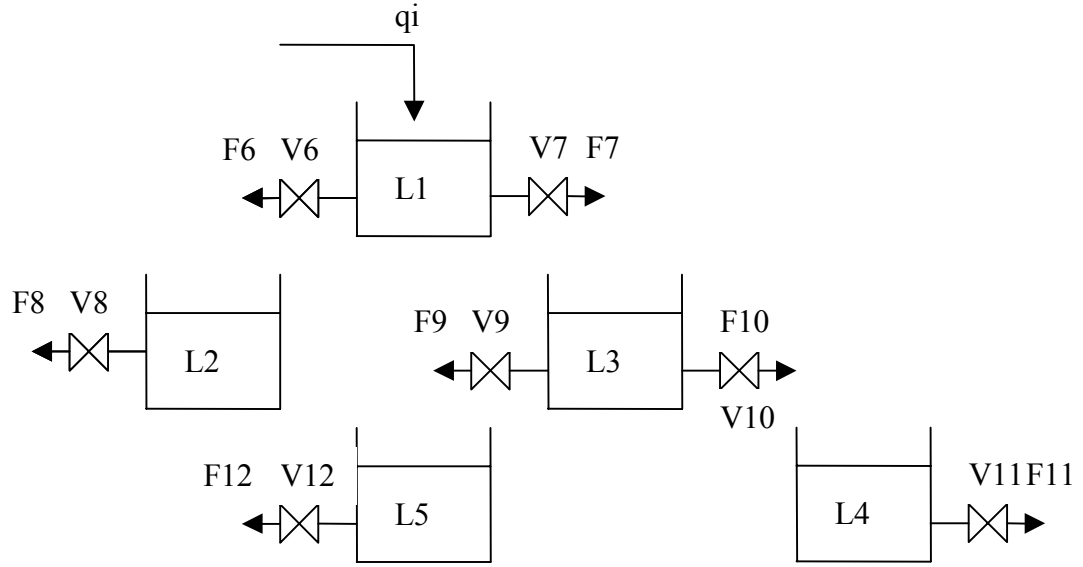


Figure 24: A five-tank system.

Thirteen faults are to be identified that are listed in Table 11. The objective is to optimally place sensors to maximize the fault detectability and minimize the number of sensors.

To simplify the case, the following assumptions are made:

- No two faults occur simultaneously;
- The fault detection sensitivity for each fault at the first sensor that the fault affects is 1 and the time-to-detection at this sensor is 1 minute;
- The detectability for each fault shares the same weight: 1;
- The SNRs of all sensor nodes are the same: 20dB;
- The cost for each sensor is 100 and the resource limit is 1000;
- Maximum number of sensors is 30.

Table 11: Faults & sensors.

Set	Fault	Sensor Nodes
A1	L1	[L1 ⁻ , L2 ⁻ , L3 ⁻ , L4 ⁻ , L5 ⁻ , F6 ⁻ , F7 ⁻ , F8 ⁻ , F9 ⁻ , F10 ⁻ , F11 ⁻ , F12 ⁻]
A2	L2	[L2 ⁻ , F8 ⁻]
A3	L3	[L3 ⁻ , L4 ⁻ , L5 ⁻ , F9 ⁻ , F10 ⁻ , F11 ⁻ , F12 ⁻]
A4	L4	[L4 ⁻ , F11 ⁻]
A5	L5	[L5 ⁻ , F12 ⁻]
A6	V6	[L1 ⁺ , L2 ⁻ , L3 ⁺ , L4 ⁺ , L5 ⁺ , F6 ⁻ , F7 ⁺ , F8 ⁻ , F9 ⁺ , F10 ⁺ , F11 ⁺ , F12 ⁺]
A7	V7	[L1 ⁺ , L2 ⁺ , L3 ⁻ , L4 ⁻ , L5 ⁻ , F6 ⁺ , F7 ⁻ , F8 ⁺ , F9 ⁻ , F10 ⁻ , F11 ⁻ , F12 ⁻]
A8	V8	[L2 ⁺ , F8 ⁻]
A9	V9	[L3 ⁺ , L4 ⁺ , L5 ⁻ , F9 ⁻ , F10 ⁺ , F11 ⁺ , F12 ⁻]
A10	V10	[L3 ⁺ , L4 ⁻ , L5 ⁺ , F9 ⁺ , F10 ⁻ , F11 ⁻ , F12 ⁺]
A11	V11	[L4 ⁺ , F11 ⁻]
A12	V12	[L5 ⁺ , F12 ⁻]
A13	qi ⁺	[L1 ⁺ , L2 ⁺ , L3 ⁺ , L4 ⁺ , L5 ⁺ , F6 ⁺ , F7 ⁺ , F8 ⁺ , F9 ⁺ , F10 ⁺ , F11 ⁺ , F12 ⁺]

The time-to-detection, the propagation time, and propagation gain information are summarized in Table 12. Meanwhile, required information for sensors and faults are given in Table 13 and Table 14 , respectively.

Table 12: Factors to determine the fault detectability.

Sensor		Direc- tion	Gain	PT/ SyD*	Sensor		Direc- tion	Gain	PT/ SyD
From	To				From	To			
F-L1**	L1	-	0.65	1/TTF	F-L2	L2	-	24.4	4/TTF
F-L3	L3	-	0.14	1/TTF	F-L4	L4	-	1.1	4/TTF
F-L5	L5	-	5.34	4/TTF	F-V6	F6	-	19.8	0/TTF
F-V7	F7	-	99	0/TTF	F-V8	F8	-	19.8	0/87
F-V9	F9	-	44	0/TTF	F-V10	F10	-	55	0/TTF
F-V11	F11	-	55	0/11	F-V12	F12	-	44	0/23
F-Qi+	L1	+	0.34	2/TTF					
L1	F6	+	5	0/TTF	F6	L1	-	0.03	2/TTF
F6	L2	+	0.8	12/ TTF	F7	L3	+	0.01	21/ TTF
L2	F8	+	1.25	0/TTF	F8	L2	-	0.95	4/TTF
L1	F7	+	25	0/TTF	F7	L1	-	0.04	1/TTF
L3	F9	+	36	0/TTF	F9	L3	—	0.01	1/TTF
F9	L5	+	0.2	81/ TTF	F10	L4	+	0.06	20/ TTF
L5	F12	+	5.05	0/TTF	F12	L5	—	0.23	4/TTF
L3	F10	+	45	0/TTF	F10	L3	—	0.01	1/TTF
L4	F11	+	16.1	0/TTF	F11	L4	-	0.07	4/TTF

*PT/SyD: PT stands for propagation time; SyD stands for symptom duration.

**F-: fault.

Table 13: Sensors information.

Sensor	Cost	SNR	Probability of Failure (*0.001)	Resolution
L1	100	6	1	0.01
L2	100	10	1	0.01
L3	100	10	1	0.01
L4	100	10	1	0.01
L5	100	10	1	0.01
F6	100	10	1	0.01
F7	100	10	1	0.01
F8	100	10	1	0.01
F9	100	10	1	0.01
F10	100	10	1	0.01
F11	100	10	1	0.01
F12	100	10	1	0.01

Table 14: Faults information.

Set	Fault	Occurrence Probability	Severity	Minimum Detectability/Reliability
A1	L1	1	1	0.5/0.9
A2	L2	1	1	0.5/0.9
A3	L3	1	1	0.5/0.9
A4	L4	1	1	0.5/0.9
A5	L5	1	1	0.5/0.9
A6	V6	1	1	0.5/0.9
A7	V7	1	1	0.5/0.9
A8	V8	1	1	0.5/0.9
A9	V9	1	1	0.5/0.9
A10	V10	1	1	0.5/0.9
A11	V11	1	1	0.5/0.9
A12	V12	1	1	0.5/0.9
A13	Qi ⁺	1	1	0.5/0.9

5.2.2 Results Based on QDG model

The QDG model of the five-tank system is built in Figure 25 and the detectability is calculated.

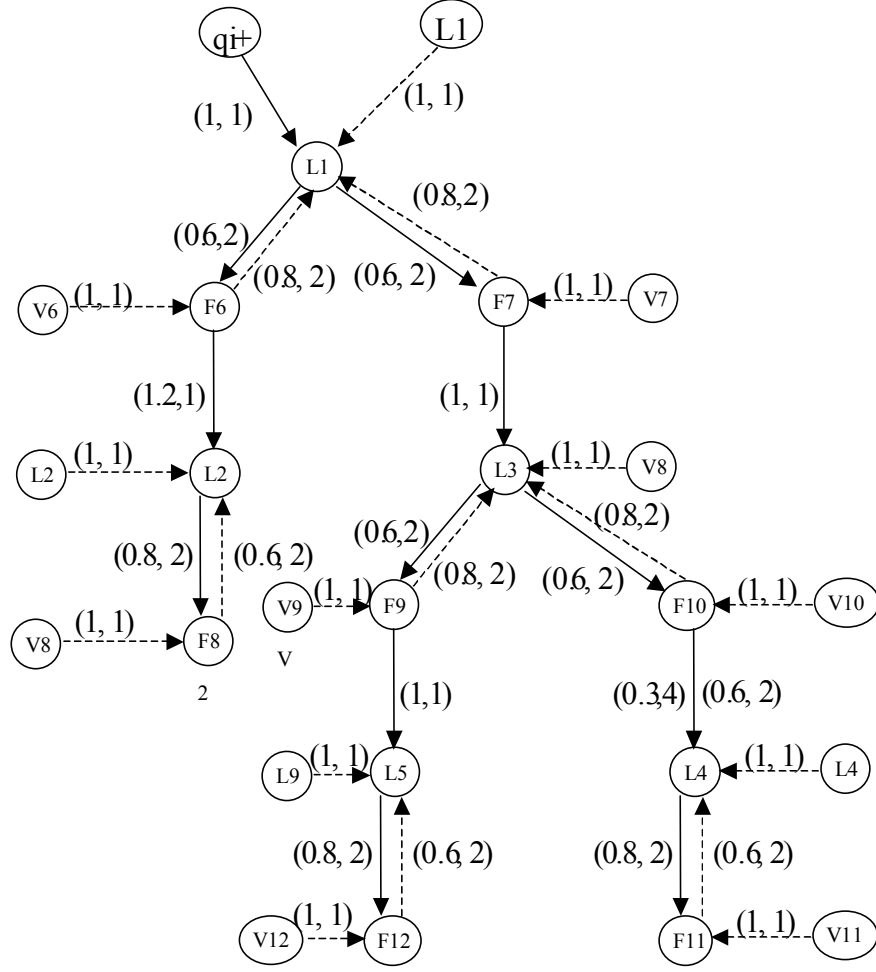


Figure 25: The QDG model of the five-tank system.

Besides the 13 original faults, an additional 55 virtual faults are created for identifying each fault. Also, since each sensor is affected by the faults in both directions, 12 virtual sensors are created and the mapping between the virtual sensors and the original sensors are built. The FOM is in the following equation:

$$\left\{ \begin{array}{l} \text{Max} \left\{ \sum_{i=1}^{68} D_i + \alpha * C_{ss} - \beta * \sum_{i=1, j=1}^{i=N, j=N} PFP_{ij} \right\} \\ \text{s.t.} \\ D_i = \frac{\sum_{j=1}^{24} b_{ij} * SD_{ij}}{\sum_{j=1}^{24} b_{ij}} \quad i = 1, 2, \dots, 68 \\ D_i > 0 \quad i = 1, 2, \dots, 68 \\ XX_j \geq 1 \Leftrightarrow \sum_{i=1}^{68} b_{ij} \geq 1 \quad j = 1, \dots, 12 \\ \alpha \geq 0 \\ \sum_{j=1}^{12} X_j + C_s = 30 \\ C_s \geq 0 \end{array} \right\}$$

The optimization results are then compared with the design obtained by using the SDG model [55] in Table 15.

Table 15: Analysis of Results.

Algorithm	# of Sensors	Sensors Selected	Minimum Detectability	# of Faults (D <= 0.5)	Average Detectability
SDG/QD G ($\alpha=1$)*	3	[L2, L4, L5]	0.31	8	0.7
QDG ($\alpha=0.5$)*	6	[L2, L4, L5, F7, F9, F10]	0.799	0	0.882

*: α is the weight of cost.

If the SDG-based approach in the appendix is used, only three sensors are selected. Because the SDG model does not contain any quantitative information, the detectability of some faults is low and the minimum detectability is 0.31 only, which means this fault can hardly be detected. Also, the detectability of 16 faults is lower than 0.5 and the average detectability is only 0.7.

With the QDG model, α , the weight of the cost in the objective function, can be adjusted easily according to different customer's requirements. The optimization results with different sensor number requirements by changing α are listed. With $\alpha = 0.5$, six sensors are selected and the average detectability is significantly improved to 0.8 if compared with the result based on the SDG model.

5.2.3 Results Considering Detectability Uncertainty

Uncertainty affects the sensor suite optimization results. Considering the fault V7 as an example. V7 is detected and identified by sensors S2 and S7. Assume the uncertainty associated with sensor fault detectability $SD(7,7)$ varies from 0.78 ± 0.1 to 0.78 ± 0.7 . If not considering uncertainty in the optimization, the worst-case total fault detectability decreases with the variation of sensor fault detectability uncertainty. Instead, if considering the uncertainty issue in the optimization, another sensor suite is selected as summarized in Table 16, and the total fault detectability is kept high, as shown in Figure 26.

Table 16: QDG method considering uncertainty.

Algorithm	# of Sensors	Sensors Selected	Minimum Detectability	# of Faults (D ≤ 0.5)	Average Detectability
QDG Considering Uncertainty ($\alpha=0.5$)*	6	[L2, L4, L5, F8, F9, F10]	0.727	8	0.875
QDG Without Considering Uncertainty ($\alpha=0.5$)*	6	[L2, L4, L5, F7, F9, F10]	0.03	13	0.212

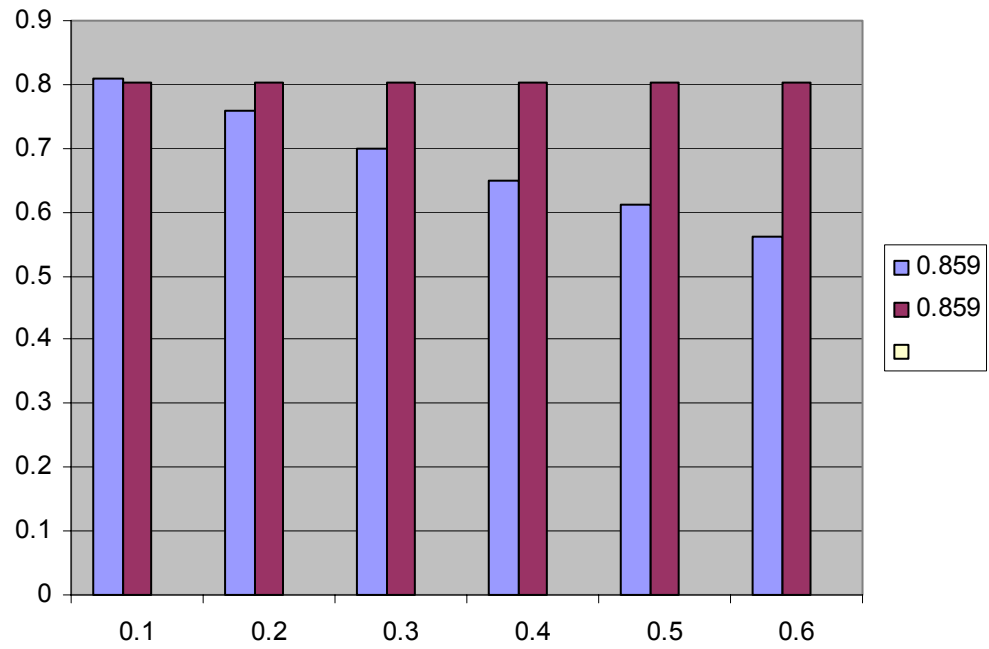


Figure 26: detectability comparison considering uncertainty

5.2.3 Performance Assessment

The optimized sensor suite is validated using the integrated diagnostic/prognostic architecture. And the performance is evaluated and compared with SDG methods (Table 17, Table 18, and Table 19).

Table 17: SDG performance analysis.

Fault	Sensors Selected	False Positive Rate (%)	False Negative Rate (%)	TTD/TTF (%)
L1	[L2, L4, L5]	7.69	0	34
L2	[L2, L4]	0	0	4.75
L3	[L2, L4, L5]	7.69	0	28
L4	[L2, L4, L5]	0	0	6.5
L5	[L2, L4, L5]	0	0	3
V6	[L2, L4]	7.69	0	21.2
V7	[L2, L4, L5]	7.69	0	53
V8	[L2, L4]	0	0	33.5
V9	[L2, L4, L5]	7.69	0	6
V10	[L2, L4, L5]	7.69	0	25
V11	[L2, L4, L5]	0	0	30.25
V12	[L2, L4, L5]	0	0	9.25
Qi+	[L2, L4]	7.69	7.69	> 1

Table 18: QDG performance analysis.

Fault	Sensors Selected	False Positive Rate (%)	False Negative Rate (%)	TTD/TTF (%)
L1	[L2, F7]	0	0	24.75
L2	[L2, F7]	0	0	4.75
L3	[F7, F9, F10]	0	0	10
L4	[L4, F7, F9, F10]	0	0	6.5
L5	[L5, F7, F9]	0	0	3
V6	[L2, F7]	0	0	11.25
V7	[L2, F7]	0	0	13
V8	[L2, F7]	0	0	6
V9	[F7, F9, F10]	0	0	3.75
V10	[F7, F9, F10]	0	0	3.25
V11	[L4, F7, F9, F10]	0	0	9.25
V12	[F5, F7, F9, F10]	0	0	6.25
Qi+	[L2, F7]	0	0	45.5

Table 19: Performance comparison.

Algorithm	Selected Sensor Number	Sensors Selected	AFPR (%)	AFNR (%)	Average	
					TTD2TTF	D
SDG/ QDG ($\alpha=1$)*	3	[L2, L4, L5]	4.14	0.59	0.184	0.7
QDG ($\alpha=0.5$)*	6	[L2, L4, L5, F7, F9, F10]	0	0	0.113	0.8

*: α is the weight of cost.

The Average False Positive Rate (AFPR) and Average False Negative Rate (AFNR) are greater than 0, while they are 0 if using our approach. Also, the time-to-detection is smaller with QDG method. Generally speaking, the performance of QDG method is better than SDG method.

5.2.4 Performance Assessment Feedback

From the performance assessment module, the ratios of time-to-detection to time-to-failure of faults 1, 6, 7, 8, 13 are larger than 10%. Therefore, we need to improve the diagnostic performance by re-adjusting the selected sensor suite.

One solution is to change the weights for these faults and run the particle-swarm optimization routine again. The results are summarized in the following tables.

Table 20: Results after performance feedback.

Fault	Sensors Selected	False Positive Rate (%)	False Negative Rate (%)	TTD/TTF (%)	Detectability
L1	[F6, F7]	0	0	5.75	0.97
L2	[L2, F6]	0	0	4.75	0.98
L3	[F6, F7, F9, F10]	0	0	10	0.98
L4	[L4, F6, F7, F9, F10, F11]	0	0	6.5	0.97
L5	[L5, F6, F7, F9]	0	0	3	0.97
V6	[F6, F7]	0	0	2.75	0.98
V7	[F6, F7]	0	0	3.75	0.99
V8	[L2, F6, F7]	0	0	6	0.96
V9	[F6, F7, F9, F10]	0	0	3.75	0.97
V10	[F6, F7, F9, F10]	0	0	3.25	0.98
V11	[L4, F6, F7, F9, F10]	0	0	9.25	0.98
V12	[L5, F6, F7, F9, F10]	0	0	6.25	0.97
Qi+	[F6, F7]	0	0	18	0.98

Table 21: Results comparison with performance feedback.

Algorithm	# of Sensors	Sensors Selected	Average			
			FPR (%)	FNR (%)	TTD2TTF	D
SDG/ QDG ($\alpha=1$)*	3	[L2, L4, L5]	4.14	0.59	0.184	0.7
QDG ($\alpha=0.5$)*	6	[L2, L4, L5, F7, F9, F10]	0	0	0.113	0.8
QDG (Performance Feedback)	7	[L2, L4, L5, F6, F7, F9, F10]	0	0	0.064	0.992

*: α is the weight of cost.

We can see from the above tables that the time-to-detection to time-to-failure ratio decreases after reselecting the sensors according to the performance feedback information. Also, the average detectability increases with the readjustment of the weights.

CHAPTER 6

CONCLUSIONS & CONTRIBUTIONS

6.1 Conclusions

The performance of PHM/CBM systems relies on the diagnostic/prognostic techniques used and the sensor suite selected. Although many sensor localization/selection, diagnostic and prognostic techniques exist in the real world, most of them were studied independently without an integrated architecture. The objective of this research is to develop an open/flexible architecture to integrate the functions of sensor localization/selection, feature extraction, mode identification, diagnostics, and prognostics, with a focus on a novel sensor localization/selection methodology for fault diagnosis.

The integrated diagnostic/prognostic architecture described in Chapter 3 provides a description of the system that is able to tackle the basic issues rising from sensors, features, and faults. Developing a diagnostic/prognostic architecture is only the first step toward an engineering solution. The next step is the actual implementation in a real-world scenario. Toward that end, the software framework necessary for support of sensor localization/selection, fault diagnostics, and prognostics has been described in this dissertation. This software framework glues individual modules together using Component Object Model (COM) technology. Also, this software framework provides an event-based communication mechanism and a common database interface that make the architecture open and scalable.

In the second part of this dissertation, a complete sensor localization/selection methodology is presented for fault diagnostic purposes. In the proposed methodology, sensor localization/selection starts with the requirements analysis; and then studies Failure Modes and Effects Criticality Analysis (FMECA), which specifies the failure modes and their criticality and probability of occurrence; quantitatively model fault propagation and calculate the sensor fault detectability and total fault detectability; formularizes and optimizes the Figure-of-Merit; and finally analyzes the performance of the diagnostic/prognostic algorithms and accordingly adjusts the sensors selected.

6.2 Significant Contributions

The significant contributions of this research include:

- A generalized methodology for sensor localization/selection for fault diagnosis.
- A quantitative definition of fault detection ability of a sensor; a novel Quantified Directed Model (QDG) method to model for the fault propagation modeling purpose; and a generalized Figure-of-Merit to maximize fault detectability and minimize the required number of sensors while achieving optimum sensor localization/selection at the system level;
- A novel, integrated architecture for diagnostic/prognostic system with an operating mode/usage pattern identifier to adapt to various industrial environments;
- Validation and Verification (V&V) of the proposed sensor localization/selection approach in the integrated diagnostic/prognostic architecture;
- The implementation of the integrated architecture under COM/DCOM structure.

6.3 Future Work

Potential for extending the research in this thesis exists in the following:

- Automate the system configuration process in the diagnostic/prognostic architecture: include feature selection, system initial configuration and reconfiguration. Currently, the features are selected manually. Feature selection algorithms are not integrated in this architecture. Meanwhile, system configuration and on-line reconfiguration are not fully supported.
- Validation of the sensor localization/selection using a physical system. The proposed methodology has been validated using a simulation system. It is necessary to test and validate the methodology using a real application.

.

REFERENCES

- [1] L. Faulds Anthony and B. King Belinda, "Sensor location in feedback control of partial differential equation systems," in Proceedings of the 2000 IEEE International Conference on Control Applications, Sept., 2000, vol. 1, pp. 536-541.
- [2] A. G. Al-Shehabi and Brett Newman, "Aeroelastic vehicle optimal sensor placement for feedback control applications using mixed gain-phase stability," in Proceedings of the American Control Conference, June 2001, vol. 3, pp. 1848-1852.
- [3] Sharon L. Padula, Rex K. Kincaid, Optimization strategies for sensor and actuator placement," NASA/TM-1999-209126, April 1999, pp. 17.
- [4] Changting Wang, Robert X. Gao, "Sensor placement strategy for In-Situ bearing defect detection," in Proceedings of the 17th IEEE, vol. 3, 2000, pp. 1463-1467.
- [5] R. Naimimohasses, D. M. Barnett, D. A. Green and P. R. Smith, "Sensor optimization using neural network sensitivity measures," Measurement Science & Technology., vol. 6, pp.1291-1300, 1995.
- [6] Furuya, H. and Haftka, R.T., "Combining Genetic and Deterministic Algorithms for Locating Actuators on Space Structures," Journal of Spacecraft and Rockets, vol. 33, no. 3, pp. 422-427, 1996.
- [7] Ponslet, E., Haftka, R.T. and Cudney, H.H., "Optimal Placement of Tuning Masses on Truss Structures by Genetic Algorithm," Collection of Technical Papers - AIAA/ASME Structures, Dynamics and Materials Conference, Part 4, pp. 2448-2457, 1993.
- [8] Yonghua Xu and Jin Jiang, "Optimal sensor location in closed-loop control systems for fault detection and isolation," in Proceedings of the American Control Conference, 2000, vol. 2, pp. 1195-1199.
- [9] R. Raghuraj, M. Bhushan, and R. Rengaswamy, "Locating sensors in complex chemical plants based on fault diagnostic observability criteria," AIChE Journal, Vol. 45, No.2, pp. 310-322, 1999.
- [10] C. Giraud and B. Jouvencel, "Sensor Selection: a Geometrical Approach," in Proceedings. 1995 IEEE/RSJ International Conference, 1995, vol. 2, pp. 555-560.
- [11] David C. Swanson, "A General Prognostic Tracking Algorithm for Predictive Maintenance," IEEE
- [12] G. Vachtsevanos and P. Wang, "An Intelligent Approach To Fault Diagnosis and Prognosis," in Proceedings on Aerospace Conference, 2001, vol. 6, pp. 2971-2977.

- [13] Pi Su, Li, Mary Nolan, Greg DeMare, and David R. Carey, "Prognostic Framework," in IEEE Systems Readiness Technology Conference on AUTOTESTCON, 1999, pp. 661-672.
- [14] S. Y. Chen and Y.F. Li, "A method of Automatic Sensor Placement for Robot Vision in Inspection Tasks," in Proceedings of the 2002 IEEE International Conference on Robotics & Automation, 2002, vol. 3, pp. 2545-2550.
- [15] Carl S. Byington, Michael J. Roemer, Thomas Galie, "Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance," in IEEE Aerospace Conference Proceedings, 2002, vol. 6, pp. 2815-2824.
- [16] Les Atlas, George Bloor, *et al.*, "An Evolvable Tri-Reasoner IVHM System," in IEEE Proceedings on Aerospace Conference, 2000, vol. 6, pp. 3023-3037.
- [17] Shashi Phoha, "Using the National Information Infrastructure (NII) for Monitoring, Diagnostics and prognostics of Operating Machinery," in Proceedings of the 35th IEEE Conference on Decision and Control, 1996, vol. 3, pp. 2583-2587.
- [18] Michael J. Roemer, Gregory J. Kacprzyński, Emmanuel O. Nwadiogbu, George Bloor, "Development of Diagnostic and Prognostic Technologies for Aerospace Health Management Applications," in IEEE Proceedings on Aerospace Conference, 2001, vol. 6, pp. 3139-3147.
- [19] Wang Haiqing, Song Zhihuan and Li Ping, "Improved PCA with Optimized Sensor Locations for Process Monitoring and Fault Diagnosis," in proceedings of the 39th IEEE Conference on Decision and Control, 2000, vol. 5, pp. 4353-4358.
- [20] Johnson, D.S., Aragon, C.R., McGeoch, L.A. and Schevon, C., "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning," Operations Research, vol. 37, 865-893, 1989.
- [21] Glover, F. And Laguna, M., "Tabu Search," Kluwer Academic Publishers, Boston, 1997.
- [22] Miguel Bagajewicz, Mabel Sanchez, "Cost-optimal design of reliable sensor networks," Computers and Chemical Engineering, vol. 23, pp. 1757-1762, 2000.
- [23] Mani Bhushan, Raghunathan Rengaswamy, "Design of Sensor Location Based on Various Fault Diagnostic Observability and reliability Criteria," Computers and Chemical Engineering, vol. 24, pp. 735-741, 2000.
- [24] M. Meyer, J. M. Lelann, B.Koehret, M. Enjalbert, "Optimal Selection of Sensor Location on a Complex Plant, Using a Graph Oriented Approach," Computers Chem. Engineering, vol. 18, pp. 8535-8540, 1994.

- [25] Donald J. Chmielewski, Tasha Palmer, and Vasilios Manousiouthakis, "On the Theory of Optimal Sensor Placement," *AIChE Journal*, vol. 48, no.5, pp.1001-1012, May 2002.
- [26] Yaqoob Ali and Shankar Narasimhan, "Sensor Network Design for Maximizing Reliability of Linear Processes," *Process System Engineering*, vol. 39, no. 5, pp. 820-828, May 1993,
- [27] Konrad, H. and Isermann, R., "Diagnosis of different faults in milling using drive signals and process models," in *Proceedings of the 13th IFAC World Congress*, vol.B., Manufacturing, p.91-96, , 1996.
- [28] Shiozaki, J., Shibata, B., Matsuyama, H., and O'shima, E., "Fault diagnosis of chemical processes utilizing signed directed graphs-improvement by using temporal information", *IEEE Transactions on Industrial Electronics*, vol.36, no.4, p.469-74, Nov. 1989.
- [29] S. Mohindra and P. A. Clark, "A Distributed Fault Diagnosis Method Based on Digraph Models: Stead-State Analysis," *Computers Chem. Engineering*, vol.17, no.2 pp.193-209, 1993.
- [30] K. B. Lim, "A Disturbance Rejection Approach to Actuator and Sensor Placement," *Journal of Guidance, Control and Dynamics*, vol. 20, no. 1, pp. 202-204, January--February 1997.
- [31] Sujoy Sen, Shankar Narasimhan and Kalyanmoy Deb, "Sensor Network Design of Linear Processes Using Genetic Algorithms," *Computers Chem. Engineering*. vol. 22 no.3, pp. 385-390, 1998
- [32] Gorinevsky, D., Dittmar, K., Mylaraswamy, D., and Nwadiogbu, E., "Model-based diagnostics for an aircraft auxiliary power unit," *Proceedings of the 2002 IEEE International Conference on Control Applications*, Sept. 18-20, 2002, Glasgow, Scotland, U.K.
- [33] Vicente, S., "Rolling bearing fault diagnostic system using fuzzy logic," 2001 IEEE International Fuzzy Systems Conference.
- [34] Filippetti, F., Granceschini, G., and Tassoni, C., "Neural networks approach to electric machine on-line diagnostics," *Fifth European Conference on Power Electronics and Applications*, Sept. 13-16, 1993, Page:213-218 vol.4
- [35] Upadhyaya, B.R., Mathai, G., and Eryurek, E., "Development and application of neural network algorithms for process diagnostics," *Proceedings of the 29th IEEE Conference on Decision and Control*, Dec. 5-7, 1990. Pages: 3277-3282, vol. 6
- [36] Zhao, L.D., and Sheng Z.H., "Combination of discrete cosine transform with neural network in fault diagnosis for rotating machinery", *Proceedings of the IEEE International Conference on Industrial Technology*, p.450-4, 2-6 Dec. 1996.

- [37] Wu, Y.S., Sun, Q., Pan, X.F., Li, X.L., Yuan, B., and Tang, X., "The application of wavelet transform and artificial neural networks in machinery fault diagnosis", Proceedings of ICSP '96 3rd International Conference on Signal Processing, p.1609-12, vol.2, 14-18 Oct. 1996.
- [38] G. Vachtsevanos and P. Wang, "A wavelet network framework for diagnostics of complex engineered systems", Maintenance and reliability Conference, MARCON 98, Knoxville, May 12-14, 1998.
- [39] P. Wang, N. Propes, N. Khiripet, Y. Li, and G. Vachtsevanos, "An integrated approach to machine fault diagnosis," IEEE Annual Textile Fiber and Film Industry Technical Conference, Atlanta, May 4-6, 1999.
- [40] P. Wang and G. Vachtsevanos, "Fault prognosis using dynamic wavelet neural networks," AAAI Symposium, Polo, Alto, March 22-24, 1999.
- [41] Ray, A. and Tangirala, S., 1996. Stochastic modeling of fatigue crack dynamics for on-line failure prognostics, IEEE Transactions on control Systems Technology, vol. 4, no. 4, p. 443-51.
- [42] Lembessis, e., antonopoulos, G., King, R.E., Halatsis, C., and torres, J., 1989. CASSANDRA": an on-line expert system for fault prognosis", Proceedings of the 5th CIM Europe Conference on Computer Integrated Manufacturing, p. 3717, 17-19.
- [43] Parker, B.E., Jr., Nigor, T.M., Carley, M.P., Barron, R.L., Ward, D.G., Poor, H.V., Rock, D., and DuBois, T.A., 1993. Helicopter gearbox diagnostics and prognostics using vibration signature analysis, Proceedings of the SPIE – The International Society for Optical Engineering, vol. 1965, p. 531-42.
- [44] Hadden, G., Vachtsevanos, G., Bennett, B. and Van Dyke, J., 1999. Machinery Diagnostics and Prognostics/Condition Based Maintenance: A Progress Report, Failure Analysis: A Foundation for Diagnostics and Prognostics Development, Proceedings of the 53rd Meeting of the Society for Machinery Failure Prevention Technology.
- [45] S.H. Cheung, K.H. Wu, and W.S. Chan, Simultaneous prediction intervals for autoregressive-integrated moving-average models: a comparative study Computational Statistics & Data Analysis, vol. 28, no. 3, pp. 297-306, Sep, 1998.
- [46] E.S.Jr. Gardner, A simple method of computing prediction intervals for time series forecasts Management Science, vol. 34, no. 4, pp. 541-6, Apr, 1998.
- [47] G. Masarotto, Bootstrap prediction intervals for autoregressions International Journal of Forecasting, vol. 6, no. 2, pp. 229-39, Jul, 1990.
- [48] R. Sharda, Neural network for the MS/OR analyst: An application bibliography , Interfaces, vol. 24, no. 2 pp. 116-130, 1994.

- [49] A.S. Weigend and N.A. Gershenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Reading, M.A. Addison-Wesley, 1993.
- [50] P.J. Werbos, Generalization of backpropagation with application to recurrent gas market model Neural Networks, vol. 1, pp. 339-356, 1988.
- [51] N. Khiripet, "An architecture for intelligent time series prediction with causal information", Ph.D. Thesis, Georgia Tech, July, 2001
- [52] P.J. Werbos, Generalization of backpropagation with application to recurrent gas market model Neural Networks, vol. 1, pp. 339-356, 1988.
- [53] Kramer, M. A.; Palowitch, B. L. J. A rule-based approach to fault diagnosis using the signed directed graph. AIChE J. 1987, 33 (7), 1067.
- [54] Mylaraswamy, D.; Kavuri, S. N.; Venkatasubramanian, V. Systematic development of causal digraph models for chemical processes part I: general framework. In AIChE Meeting, San Francisco, CA, 1994.
- [55] Bhushan, M. and Rengaswamy, R. Design of sensor network based on the signed directed graph of the process for efficient fault diagnosis
- [56] Chang, C. T.; Mah, K. N.; Tsai, C. S. A simple design strategy for fault monitoring systems. AIChE J. 1993, 39 (7), 1146.
- [57] R. Saeks and J. Pooley, "A hybrid supervised/unsupervised neural network architecture for health and usage monitoring," IEEE, 1998
- [58] N., Khiripet, "An architecture for intelligent time series prediction with causal information," Ph.D. Thesis, Georgia Institute of Technology, May 2001.
- [59] P. Wang, "Intelligent signal/image processing for fault diagnosis and prognosis," Ph.D. Thesis, Georgia Institute of Technology, February 2000.
- [60] K. Keller, D. Wiegand, K. Swearingen, C. Reisig, S. Black, A. Gillis, M. Vandernoot, "An architecture to implement integrated vehicle health management systems," 2001 IEEE
- [61] Seungkoo Lee, "An architecture for a diagnostic/prognostic system with rough set feature selection and diagnostic decision fusion capabilities," Ph.D. Thesis, Georgia Tech, December 2002.
- [62] N. Propes, "Hybrid system diagnostics and control reconfiguration for manufacturing systems," PhD Proposal, School of ECE at Georgia Tech, May, 2002.
- [63] G. Zhang, S. Lee, N. Propes, Y. Zhao, G. Vachtsevanos, A. Thakker, and T. Galie, "A novel architecture for an integrated fault diagnostic/prognostic system," AAAI Symposium, Stanford, California, March 25-27, 2002.

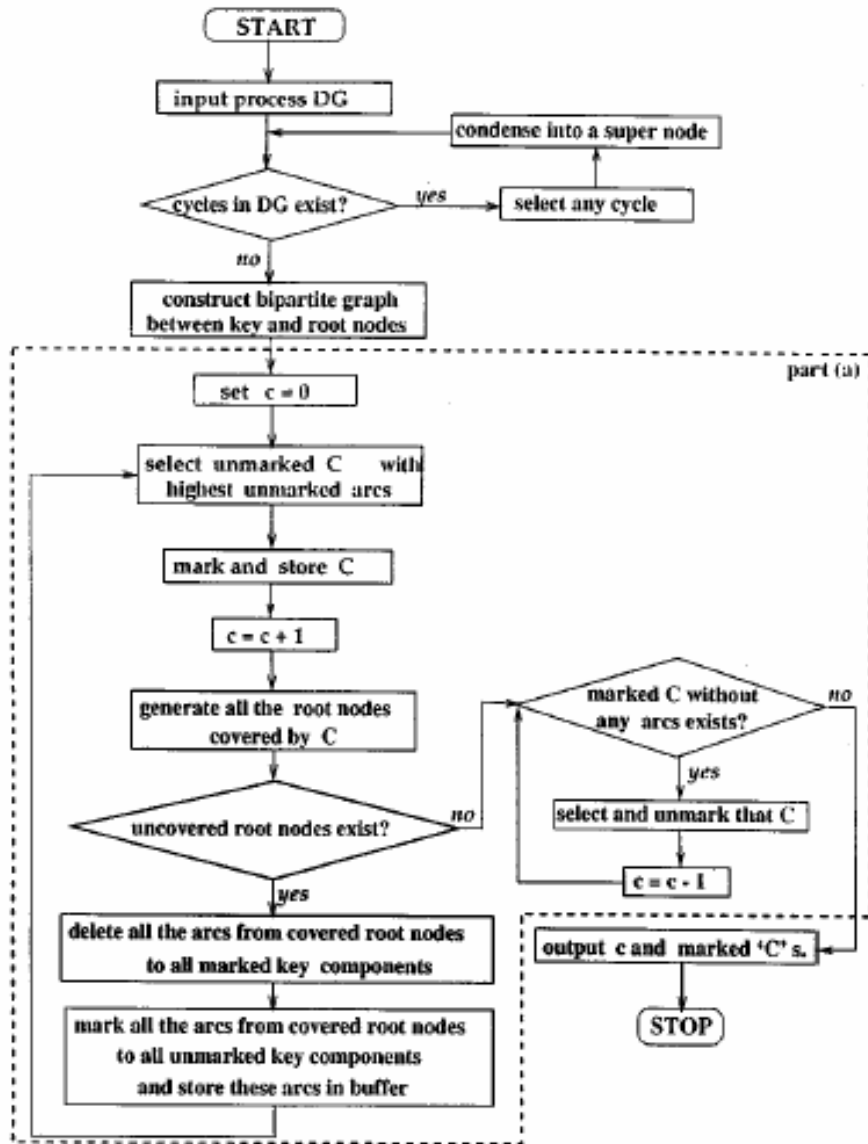
- [64] Eberhart, R. C. and Kennedy, J, "A new optimization using particle swarm theory," Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39-43, 1995.
- [65] Y. Zhang, "Product quality modeling & control based on vision inspection with an application to baking processes," Ph.D. Thesis in preparation, Georgia Tech, December 2004.
- [66] B. Wu, G. Zhang, G. Vachtsevanos, "Prognostic enhanced diagnostics: verification and validation." Paper in preparation.
- [67] Anthony J. Wheeler, Ahmad R. Ganji, "Introduction to Engineering Experimentation," Englewood cliffs, New Jersey, Prentice Hall, 1996
- [68] M. G. Morgan and M. Henrion. Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and fblcity Analysis. Cambridge Univ. Press, 1990.
- [69] Chen, W., Baghdasaryan, L., Buranathiti, T., Cao, J., 2004, Model validation via uncertainty propagation and data transformations, AIAA Journal, 42, 7, 1406-1415.
- [70] Sastry S. Isukapalli, "Uncertainty Analysis of Transport-Transformation Models," Ph.D. dissertation, the state university of New Jersey, 1999.
- [71] Agrawal KN, Kumar A, Behabadi MAA and Varma HK, (1998), Heat transfer augmentation by coiled wire inserts during forced convection condensation of R22 inside horizontal tubes, International journal of multiphase flow, Vol. 24, pp. 635-650
- [72] Bremser W, (2001), Measurement uncertainty in testing exemplified: an application to gas metering, Test and measurement 2001 workshop / conference, September 9 – 12 2001, South Africa
- [73] Dougherty EP and Rabitz H, (1979), A computational algorithm for the Green's function method of sensitivity analysis in chemical kinetics, International journal of chemical kinetics, Vol. 11, pp. 1237-1249
- [74] Dougherty EP , Hwang JT and Rabitz H, (1979), Further developments and applications of the green's function method of sensitivity analysis in chemical kinetics, International journal of chemical kinetics , Vol. 71 (4), pp. 1794-1808
- [75] Dunker AM, (1984), The decoupled direct method for calculating sensitivity coefficients in chemical kinetics, Journal of chemical physics , Vol. 51(5), pp. 2385-2393
- [76] Hodges James S, (1987), Uncertainty, policy analysis and statistics, Statistical science, Vol. 2, no. 3, pp. 259-291

- [77] Iman RL and Conover WJ, (1980), Small sample sensitivity analysis techniques for computational models with an application to risk assessment, Communications in statistics part a - theory and methods, Vol. 17, pp. 1749-1842
- [78] Kline SJ and McClintock FA, (1953), Describing uncertainties in single-sample experiments, Mechanical engineering , Vol. 75, pp. 3-8
- [79] Koley NI and Hofer E, (1996), Uncertainty and sensitivity analysis of a postexperiment simulation of non-explosive melt-water interaction, Experimental thermal and fluid science, Vol. 13, pp. 98-116
- [80] Loh WL, (1996), On Latin hypercube sampling, Annals of statistics, Vol. 24(5), pp. 2058-2080
- [81] Mosteller F and Youtz C, (1990), Quantifying probabilistic expressions, Statistical Science; Vol. 5; pp. 1-34
- [82] Müller N, (2001), Introducing the concept of uncertainty of measurement in testing in association with the application of the standard ISO/IEC 17025, Test and measurement 2001 workshop / conference, September 9 – 12 2001, South Africa
- [83] Naidoo K, (2001), Identifying and dealing with measurement uncertainty in the process industry, Test and measurement 2001 workshop / conference , September 9 – 12 2001, South Africa
- [84] Pearson K, (1902), On the mathematical theory of errors in judgement with special reference to the personal equation, Philosophical transactions of the royal society of London, series A, Vol. 198, pp. 235-299
- [85] Prokopakis GJ, (1993), The decoupled direct method for calculating sensitivity coefficients in chemical kinetics, Applied mathematical modelling, Vol. 17(9), pp. 499-503
- [86] Schultz RR and Cole R, (1979), Uncertainty analysis in boiling nucleation, AIChE Symposium Sr., Vol. 75, no. 189, pp. 32-38
- [87] Schultz RR, Kasturirangan S and Cole R, (1975), Experimental studies of incipient vapour nucleation, The Canadian journal of chemical engineering, Vol. 35, pp. 408-413
- [88] Shafer G, (1987), Probability judgement in artificial intelligence and expert systems, Statistical science , Vol. 2, pp. 1-44
- [89] Stein M, (1987), Large sample properties of simulations using latin hypercube sampling, Technometrics , Vol. 29(2), pp. 143-151
- [90] Tavener JP, (2001), Uncertainties: a bar to progress, Test and measurement 2001 workshop / conference, September 9 – 12 2001, South Africa

- [91] Barrett TS, (1996), Probabilistic design for rotordynam in simulations using receptance -based re-analysis, PhD thesis, Texas A&M University
- [92] Coetzee S, (2000), The development of an experimental set-up to investigate heat transfer enhancement in tube-in-tube heat exchangers, MEng thesis, Rand Afrikaans University
- [93] Dirker J, (1999), Heat transfer augmentation during condensation of r-22 with two spiralled wires in the annulus of a coiled tube -in-tube heat exchanger for hot water heat pumps, BSc thesis, Rand Afrikaans University
- [94] Tatang MA, (1992), Combined stoc hastic and deterministic approach in solving stochastic differential equations, Master's thesis Carnegie Mellon university
- [95] Tatang MA, (1995), Direct incorporation of uncertainty in chemical and environmental engineering systems, PhD thesis Massachusetts Institute of Technology

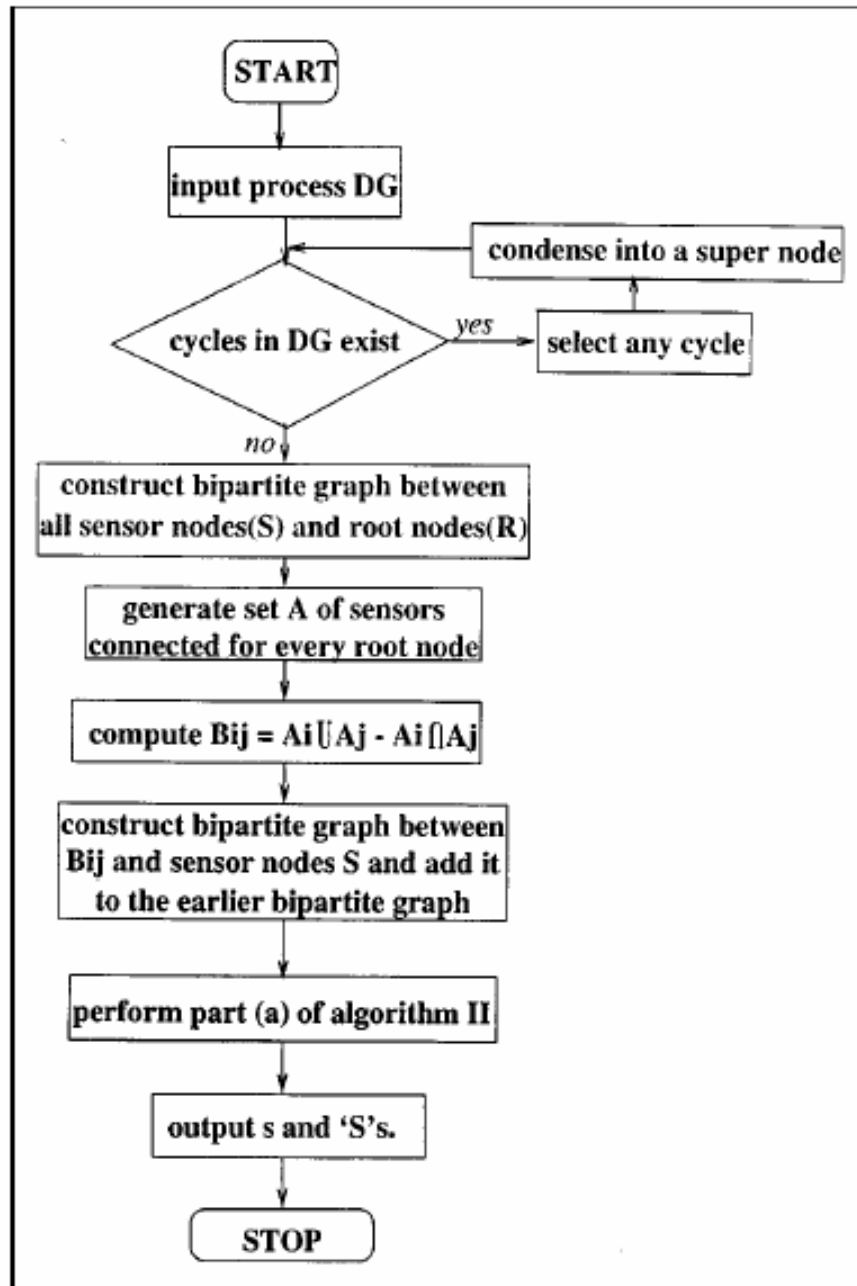
APPENDIX A

FLOW CHART FOR GREEDY SEARCH ALGORITHMS II TO SOLVE OBSERVABILITY [9]



APPENDIX B

FLOW CHART FOR GREEDY SEARCH ALGORITHM III TO ACHIEVE MAXIMUM FAULT RESOLUTION [9]



APPENDIX C

LIST OF PUBLICATIONS

- **G. Zhang**, S. Lee, N. Propes, Y. Zhao, G. Vachtsevanos, A. Thakker, and T. Galie, "A Novel Architecture for an Integrated Fault Diagnostic/Prognostic System," AAAI Symposium, Stanford, California, March 25-27, 2002.
- N. Propes, S. Lee, **G. Zhang**, I. Barlas, Y. Zhao, G. Vachtsevanos, A. Thakker, and T. Galie, "A Real-Time Architecture for Prognostic Enhancements to Diagnostic Systems," MARCON, Knoxville, Tennessee, May 6-8, 2002.
- I. Barlas, **G. Zhang**, N. Propes, G. Vachtsevanos, T. Galie, and A. Thakkar, "Addressing Uncertainty and Confidence in Prognosis," Intelligent Ship Symposium, Philadelphia, May, 2003.
- I. Barlas, **G. Zhang**, N. Propes, and G. Vachtsevanos, "Confidence Metrics and Uncertainty Management in Prognosis," MARCON, Knoxville, May 2003.
- Seungkoo Lee, Nicholas Propes, **Guangfan Zhang**, Yongshen Zhao, George J. Vachtsevanos: Rough Set Feature Selection and Diagnostic Rule Generation for Industrial Applications. Rough Sets and Current Trends in Computing 2002: 568-571

VITA

Guangfan Zhang was born in P. R. China in April 1975. He received both of his Bachelor of Engineering degree and Ph. D. degree in Automation from Shanghai Jiao Tong University in 1994 and 1999, respectively. He joined the Intelligent Control Systems Lab in School of Electrical and Computer Engineering, Georgia Institute of Technology in 1999, where he started to work as a research assistant under the supervision of Dr. George. J. Vachtsevanos. He received his Master of Science degree in 2002 and Ph.D. degree in 2005 in Electrical and Computer Engineering from Georgia Institute of Technology. His research interest includes sensor selection/localization and fault diagnostics/prognostics for condition based maintenance, and systems & control.ization and fault diagnostics/prognostics for condition based maintenance, and systems & control.